

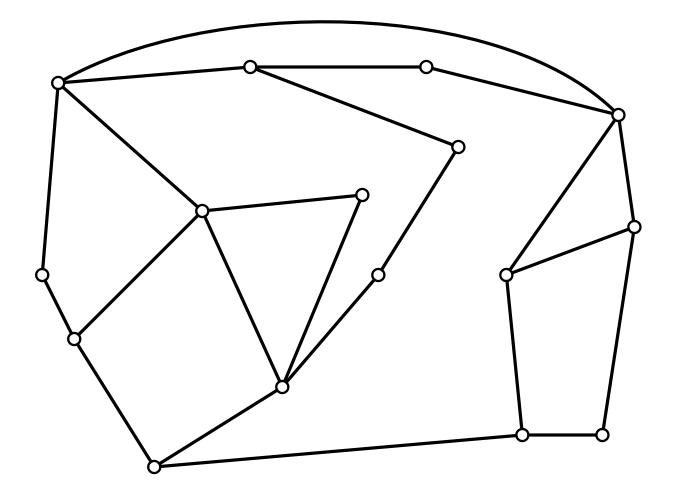


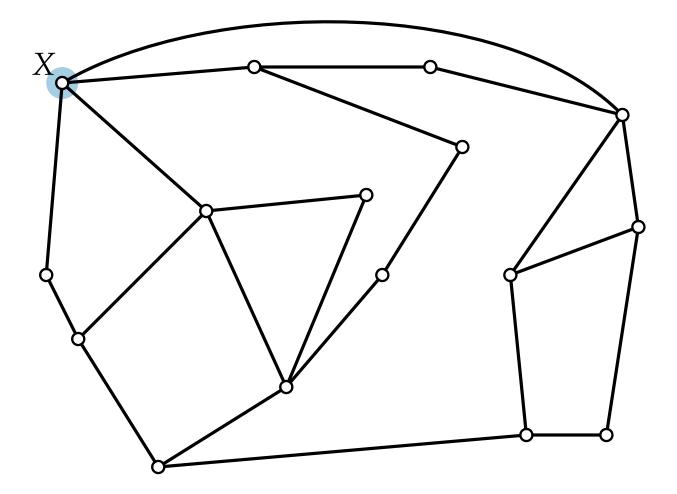
Finding Tutte Paths in Linear Time

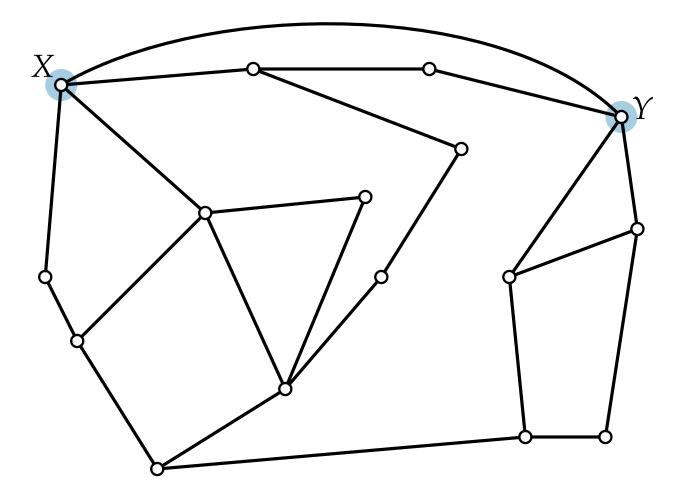
Philipp Kindermann Universität Würzburg

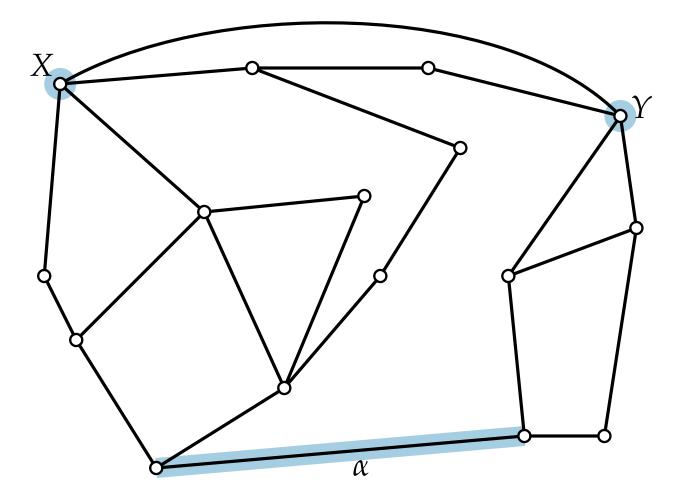
joint work with Therese Biedl University of Waterloo

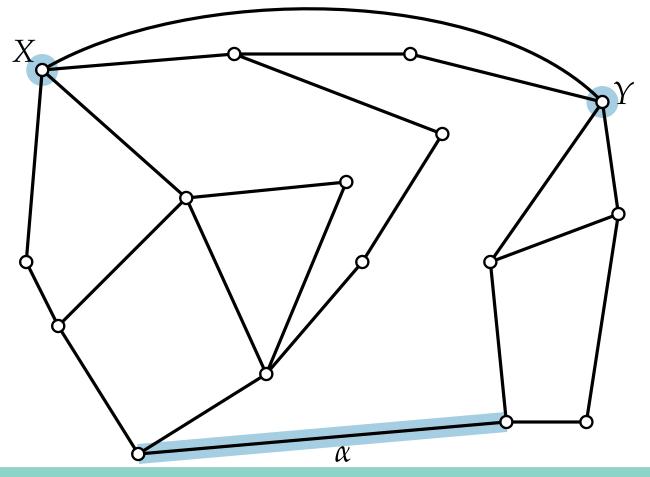




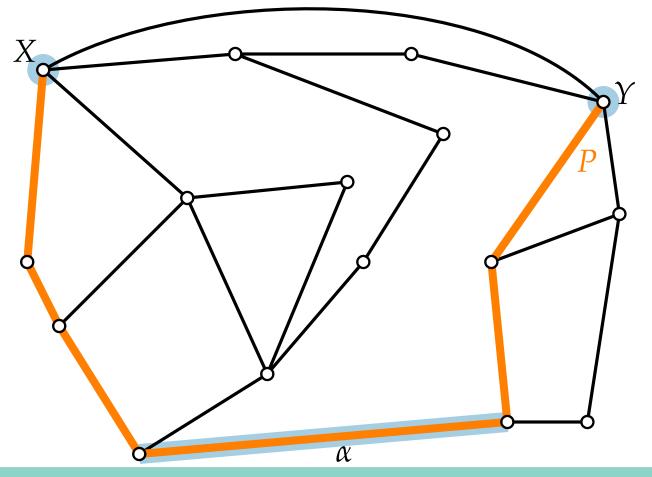




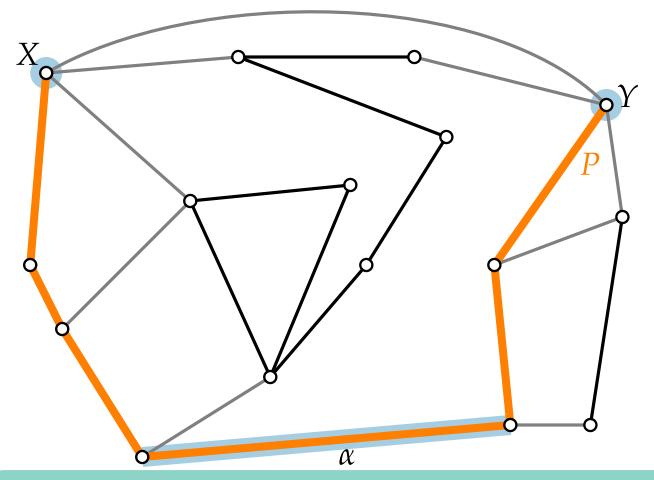




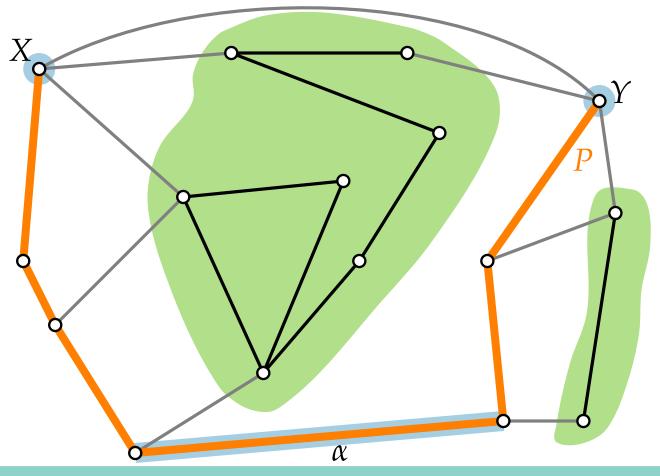
Planar graph G



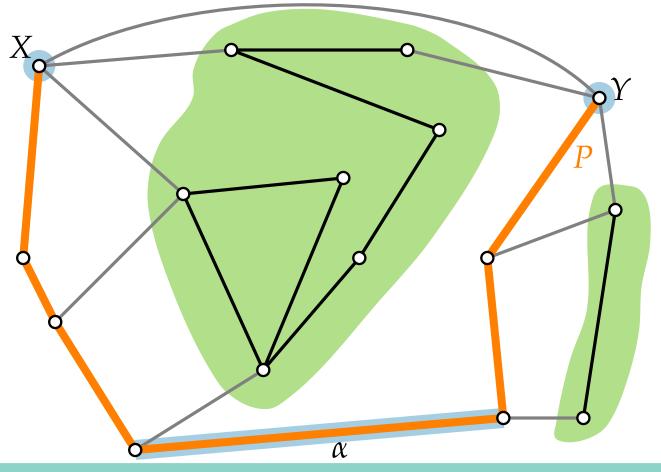
Planar graph G



Planar graph G



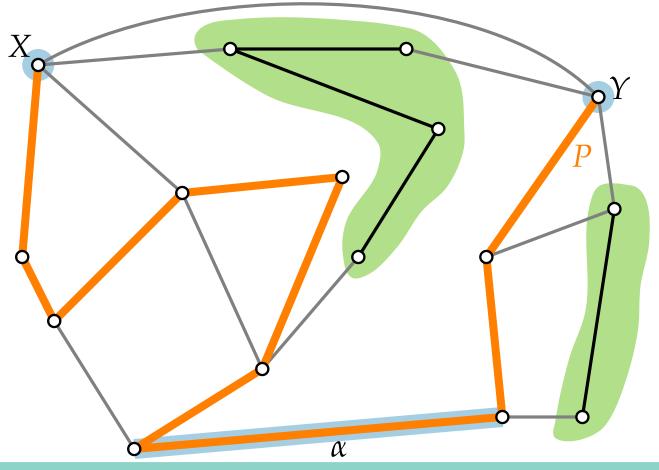
Planar graph G



Planar graph G

Tutte path: Path from X to Y via α

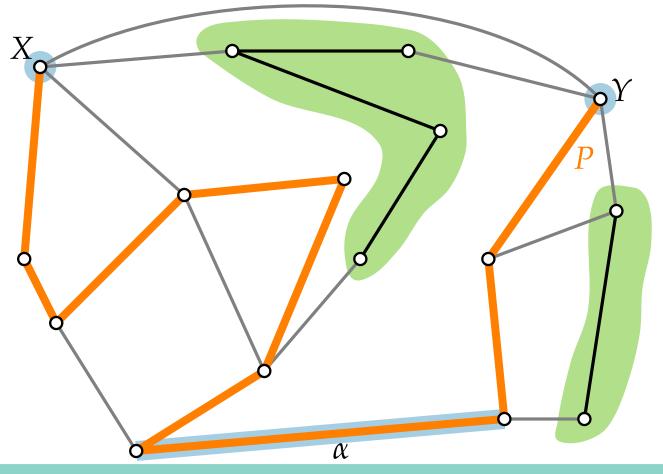
Every comp. attached to ≤ 3 vtcs of P



Planar graph G

Tutte path: Path from X to Y via α

Every comp. attached to ≤ 3 vtcs of P

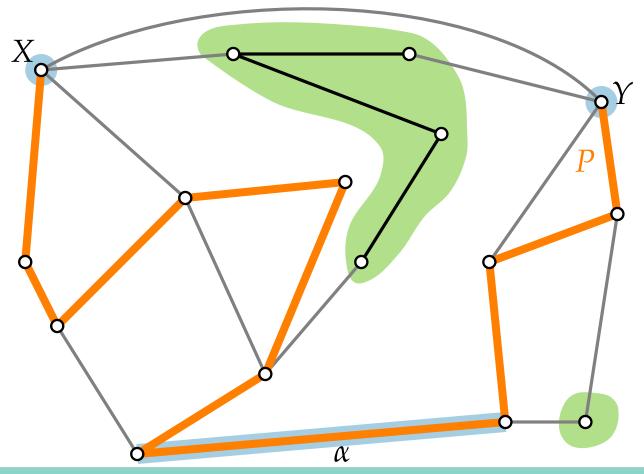


Planar graph G

Tutte path: Path from X to Y via α

Every comp. attached to ≤ 3 vtcs of P

Every outer comp. attached to 2 vtcs of *P*

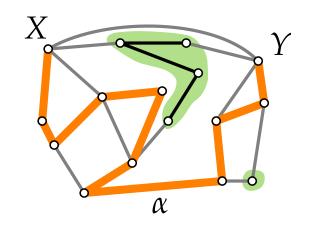


Planar graph G

Tutte path: Path from X to Y via α

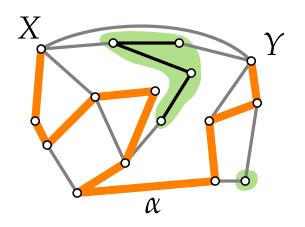
Every comp. attached to ≤ 3 vtcs of P

Every outer comp. attached to 2 vtcs of *P*



[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

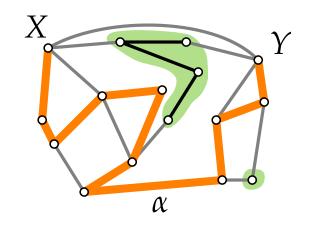


[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path



[Tutte '77]

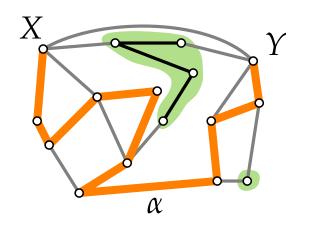
G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, X, α on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., X, X α on outer face \Rightarrow Tutte path



[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

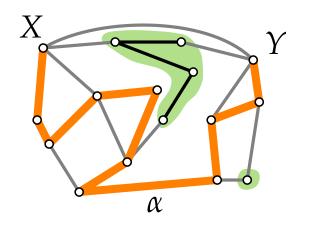
G 2-conn., X, \times α on outer face \Rightarrow Tutte path

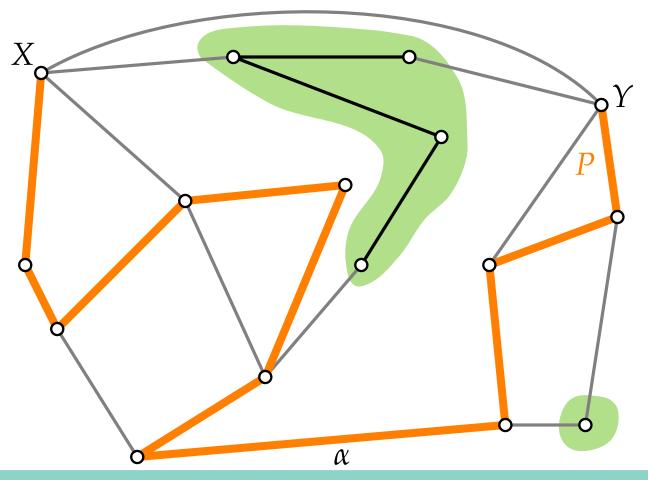
[Sanders '96]

G 2-conn., X, X α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., *X*, *Y*, α on outer face \Rightarrow T_{SDR} -path



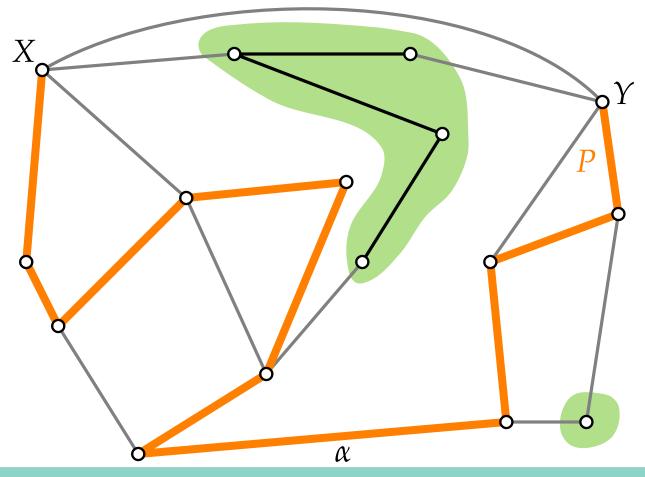


Planar graph G

Tutte path: Path from X to Y via α

Every comp. attached to ≤ 3 vtcs of P

Every outer comp. attached to 2 vtcs of *P*



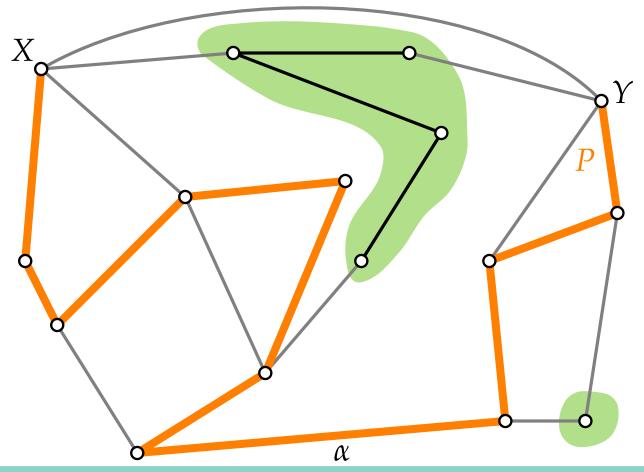
Planar graph G

Tutte path: Path from X to Y via α

Every comp. attached to ≤ 3 vtcs of P

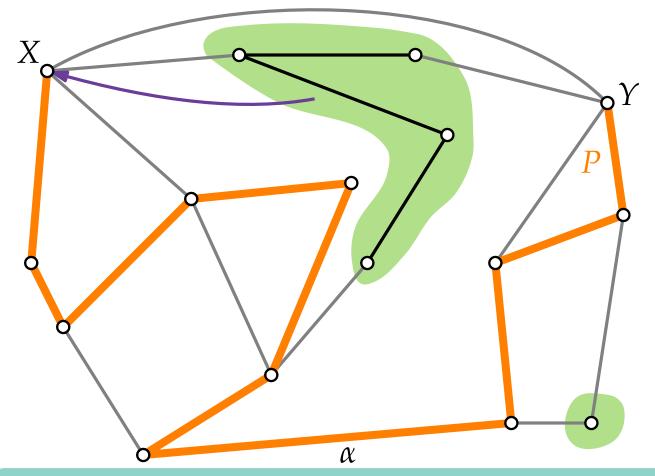
Every outer comp. attached to 2 vtcs of *P*

 $T_{\rm SDR}$ -path: Tutte path + System of Distinct Representatives:



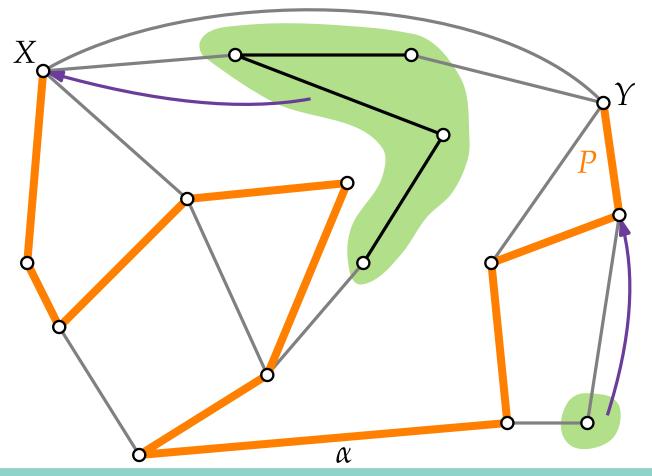
Planar graph G

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of PEvery outer comp. attached to 2 vtcs of P



Planar graph G

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of PEvery outer comp. attached to 2 vtcs of P



Planar graph G

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of PEvery outer comp. attached to 2 vtcs of P

[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

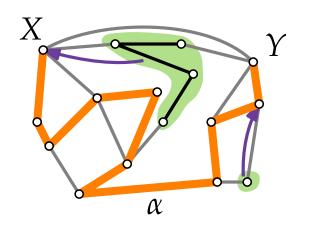
G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., X, X α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., *X*, *Y*, α on outer face \Rightarrow T_{SDR} -path



[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

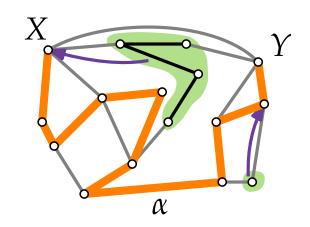
G 2-conn., X, X α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., *X*, *Y*, α on outer face \Rightarrow T_{SDR} -path

[Chiba & Nishizeki '89]

G 4-conn. \Rightarrow Tutte path in O(n) time



[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

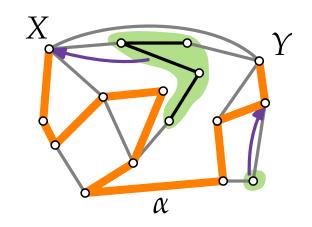
G 2-conn., X, X, α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., *X*, *Y*, α on outer face \Rightarrow T_{SDR} -path

[Chiba & Nishizeki '89] (= Hamil. path)

G 4-conn. \Rightarrow Tutte path in O(n) time



[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., X, X, α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

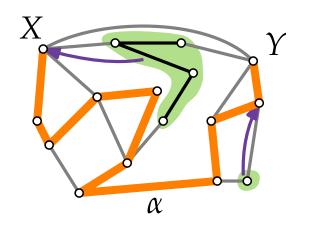
G 3-conn., *X*, *Y*, α on outer face \Rightarrow T_{SDR} -path

[Chiba & Nishizeki '89] (= Hamil. path)

G 4-conn. \Rightarrow Tutte path in O(n) time

[Schmid & Schmidt '15]

... in $O(n^2)$ time



[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, X on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., X, X, α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., *X*, *Y*, α on outer face \Rightarrow T_{SDR} -path

[Chiba & Nishizeki '89] (= Hamil. path)

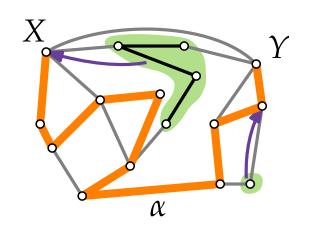
G 4-conn. \Rightarrow Tutte path in O(n) time

[Schmid & Schmidt '15]

... in $O(n^2)$ time

[Schmid & Schmidt '18]

... in $O(n^2)$ time



[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path



G 2-conn., \times \times α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., X, Y, α on outer face \Rightarrow T_{SDR} -path

[Chiba & Nishizeki '89] (= Hamil. path)

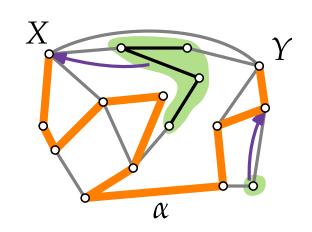
G 4-conn. \Rightarrow Tutte path in O(n) time

[Schmid & Schmidt '15]

... in $O(n^2)$ time

[Schmid & Schmidt '18]

... in $O(n^2)$ time



 \dots in O(n) time

[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., \times \times α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., *X*, *Y*, α on outer face \Rightarrow T_{SDR} -path

[Chiba & Nishizeki '89] (= Hamil. path)

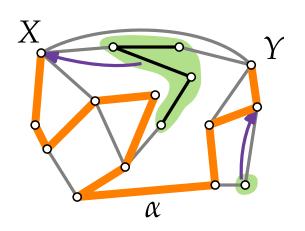
G 4-conn. \Rightarrow Tutte path in O(n) time

[Schmid & Schmidt '15]

... in $O(n^2)$ time

[Schmid & Schmidt '18]

... in $O(n^2)$ time



... in O(n) time

Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., X, X, α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., X, Y, α on outer face $\Rightarrow \mathcal{I}_{SDR}$ -path

[Chiba & Nishizeki '89] (= Hamil. path)

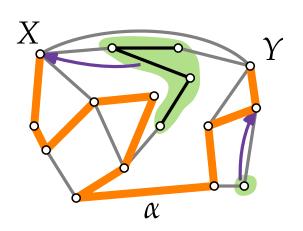
G 4-conn. \Rightarrow Tutte path in O(n) time

[Schmid & Schmidt '15]

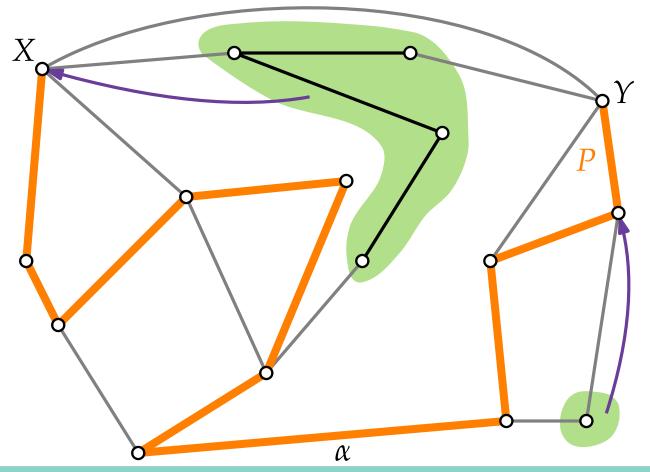
... in $O(n^2)$ time

[Schmid & Schmidt '18]

... in $O(n^2)$ time

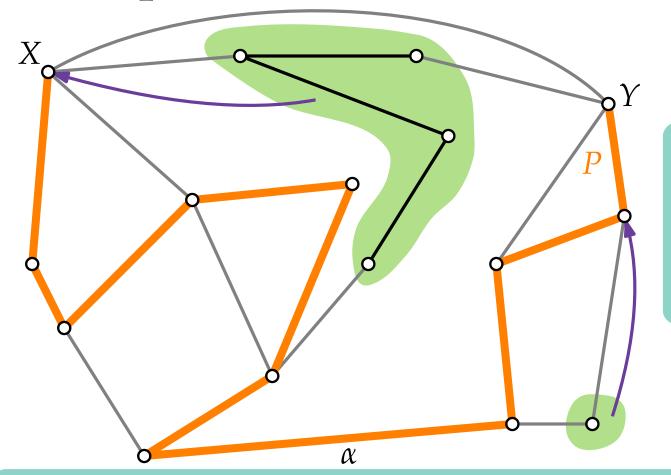


 \dots in O(n) time



Planar graph G

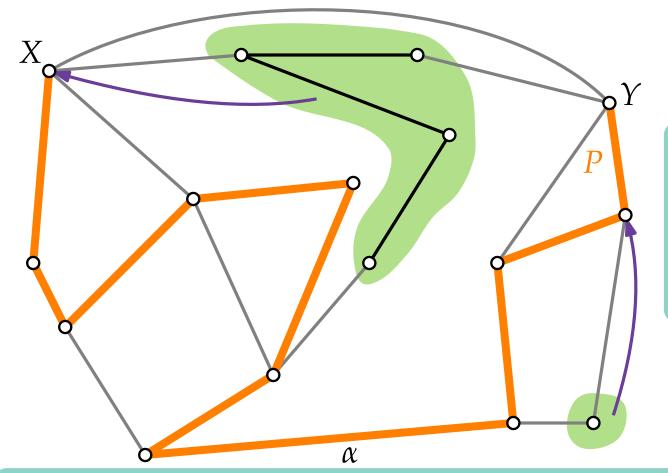
Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of PEvery outer comp. attached to 2 vtcs of P



Planar graph G

 T_{int} -path:

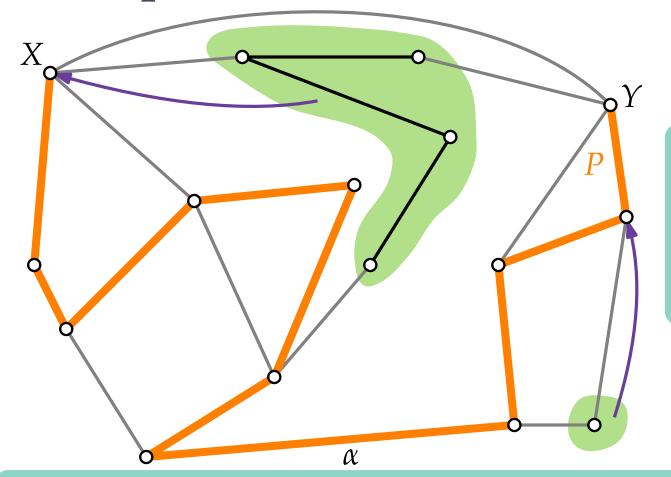
Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of PEvery outer comp. attached to 2 vtcs of P



Planar graph G

 T_{int} -path: $-T_{\text{SDR}}$ -path

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of PEvery outer comp. attached to 2 vtcs of P

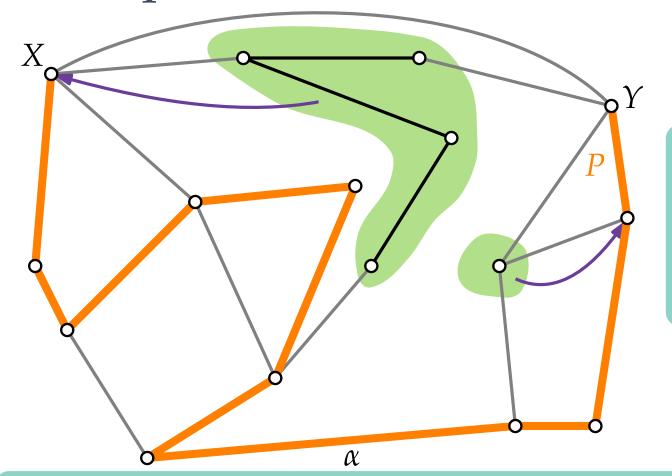


Planar graph G

*T*_{int}-path:

- $-T_{\rm SDR}$ -path
- visits all ext. vtcs

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of PEvery outer comp. attached to 2 vtcs of P



Planar graph G

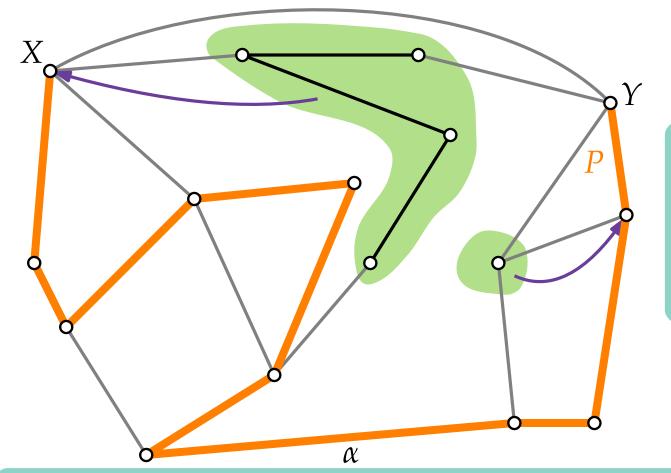
 T_{int} -path:

- $-T_{\rm SDR}$ -path
- visits all ext. vtcs

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of P

Every outer comp. attached to 2 vtcs of *P*

Tutte paths



Planar graph G

*T*_{int}-path:

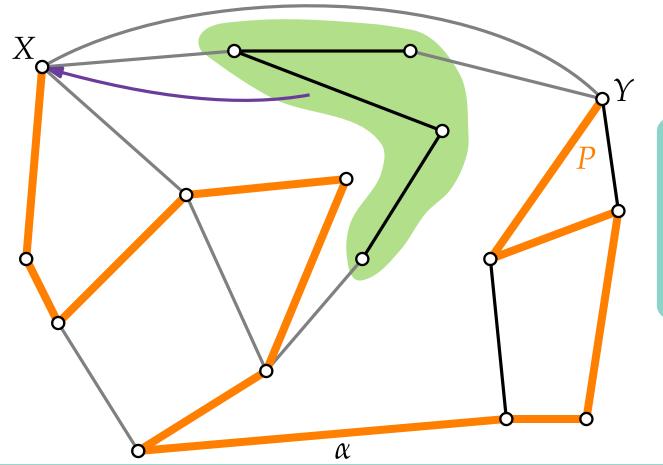
- $-T_{\rm SDR}$ -path
- visits all ext. vtcs
- all comp. assigned to int. vtcs

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of P

Every outer comp. attached to 2 vtcs of *P*

 $T_{\rm SDR}$ -path: Tutte path + System of Distinct Representatives: Injective assignment of comp. to attachment pts

Tutte paths



Planar graph G

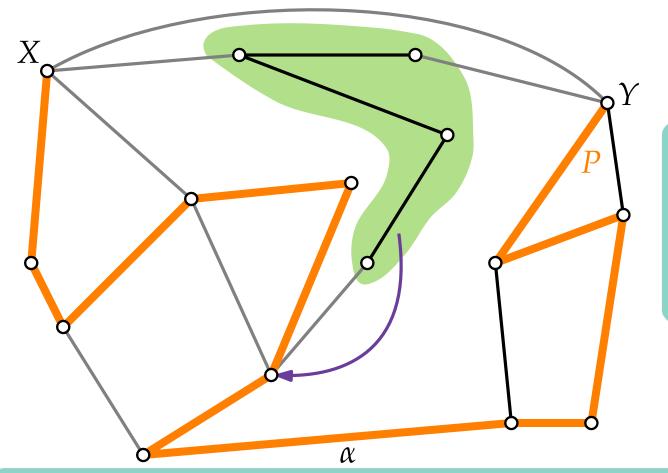
T_{int}-path:

- $-T_{\rm SDR}$ -path
- visits all ext. vtcs
- all comp. assigned to int. vtcs

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of PEvery outer comp. attached to 2 vtcs of P

 $T_{\rm SDR}$ -path: Tutte path + System of Distinct Representatives: Injective assignment of comp. to attachment pts

Tutte paths



Planar graph G

T_{int}-path:

- $-T_{\rm SDR}$ -path
- visits all ext. vtcs
- all comp. assigned to int. vtcs

Tutte path: Path from X to Y via α Every comp. attached to ≤ 3 vtcs of P

Every outer comp. attached to 2 vtcs of *P*

 $T_{\rm SDR}$ -path: Tutte path + System of Distinct Representatives: Injective assignment of comp. to attachment pts

What is known?

Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., X, X, α on outer face \Rightarrow Tutte path

[Gao, Richter & Yu '95, '06]

G 3-conn., X, Y, α on outer face $\Rightarrow \mathcal{I}_{SDR}$ -path

[Chiba & Nishizeki '89] (= Hamil. path)

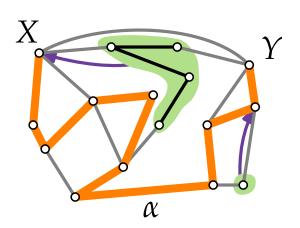
G 4-conn. \Rightarrow Tutte path in O(n) time

[Schmid & Schmidt '15]

... in $O(n^2)$ time

[Schmid & Schmidt '18]

... in $O(n^2)$ time



 \dots in O(n) time

What is known?

Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., X, X, α on outer face \Rightarrow Tutte path

int.o, Richter & Yu '95, '06]

G'3-conn., X, Y, α on outer face $\Rightarrow \mathcal{D}_{SR}$ -path

[Chiba & Nishizeki '89] (= Hamil. path)

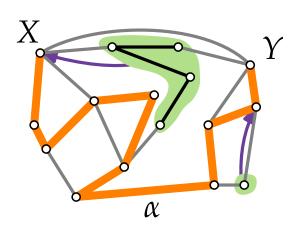
G 4-conn. \Rightarrow Tutte path in O(n) time

[Schmid & Schmidt '15]

... in $O(n^2)$ time

[Schmid & Schmidt '18]

... in $O(n^2)$ time



 \dots in O(n) time

What is known?

[Tutte '77]

G 2-conn., *X*, *Y*, α on outer face \Rightarrow Tutte path

[Thomassen '83]

G 2-conn., X, \times α on outer face \Rightarrow Tutte path

[Sanders '96]

G 2-conn., X, X, α on outer face \Rightarrow Tutte path

int.o, Richter & Yu '95, '06]

G'3-conn., X, Y, α on outer face $\Rightarrow \mathcal{D}_{SR}$ -path

[Chiba & Nishizeki '89] (= Hamil. path)

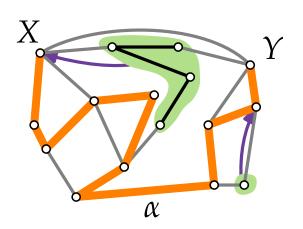
G 4-conn. \Rightarrow Tutte path in O(n) time

[Schmid & Schmidt '15]

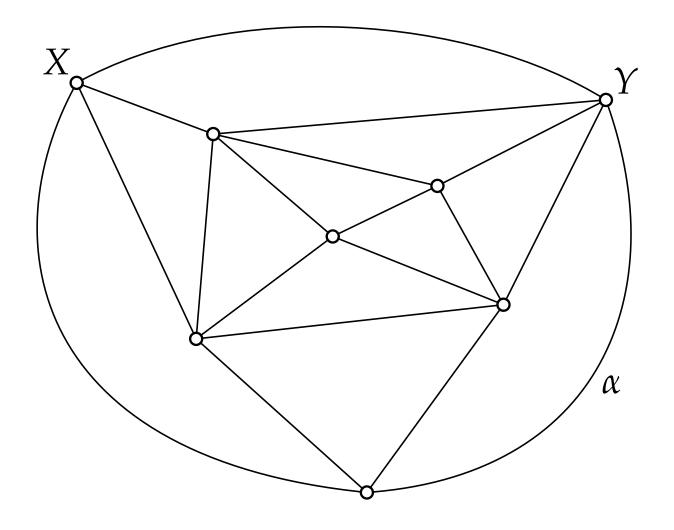
... in $O(n^2)$ time

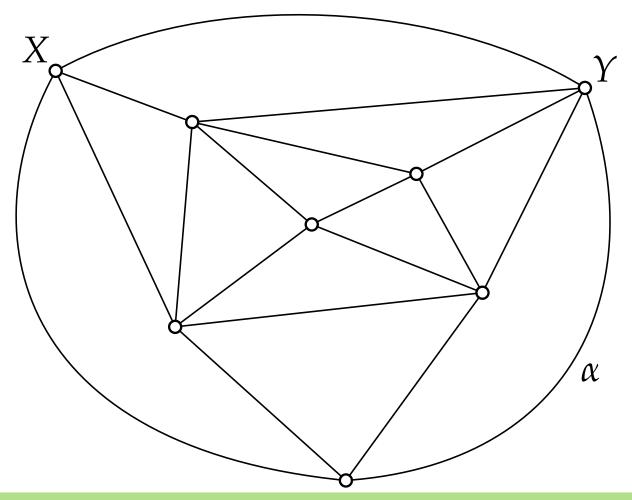
[Schmid & Schmidt '18]

... in $O(n^2)$ time

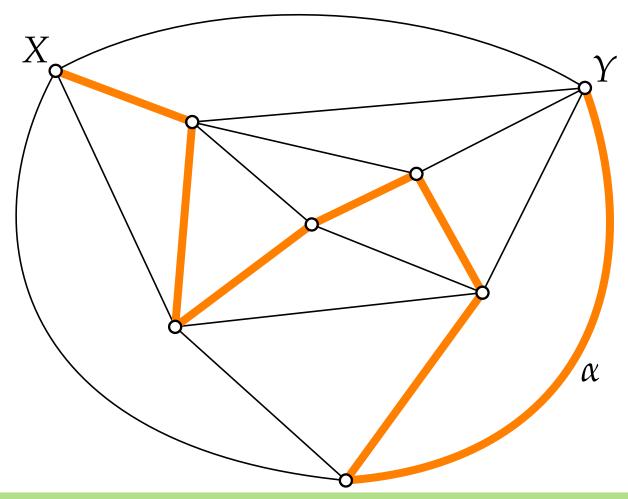


 \dots in O(n) time

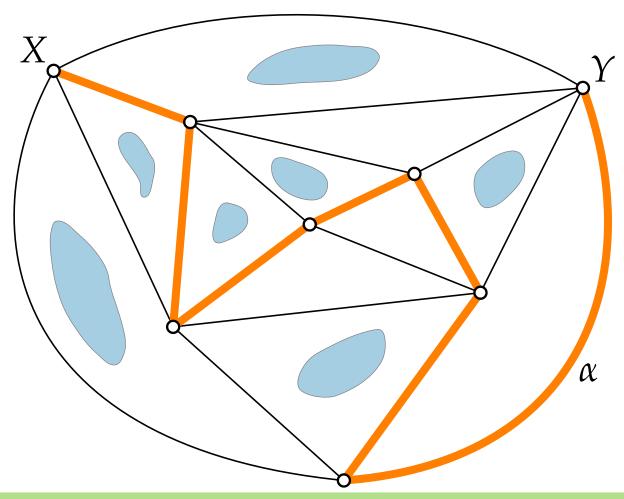




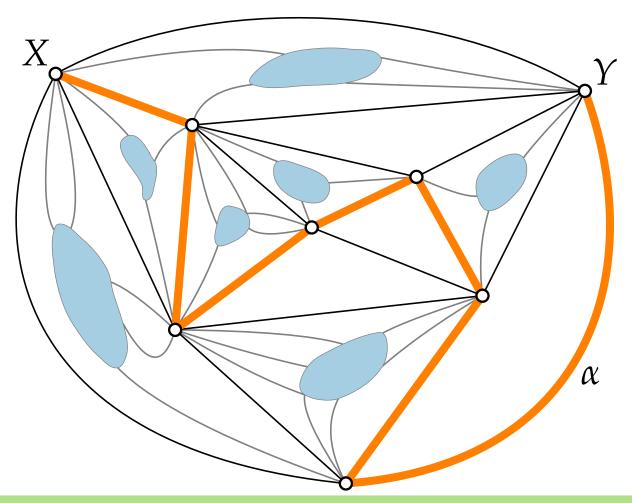
[Asano, Kikuchi & Saito '85]



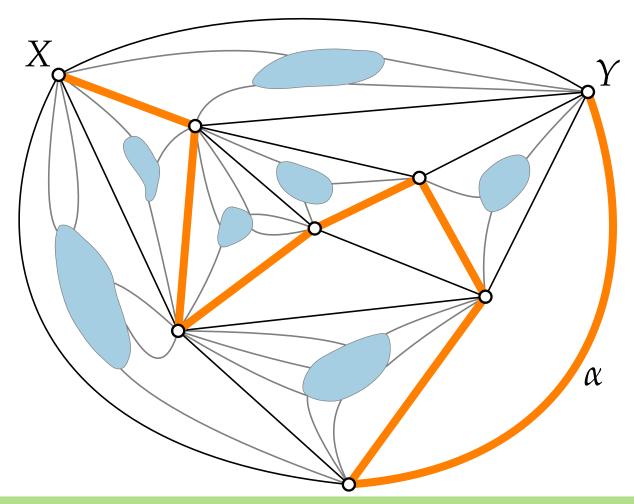
[Asano, Kikuchi & Saito '85]



[Asano, Kikuchi & Saito '85]

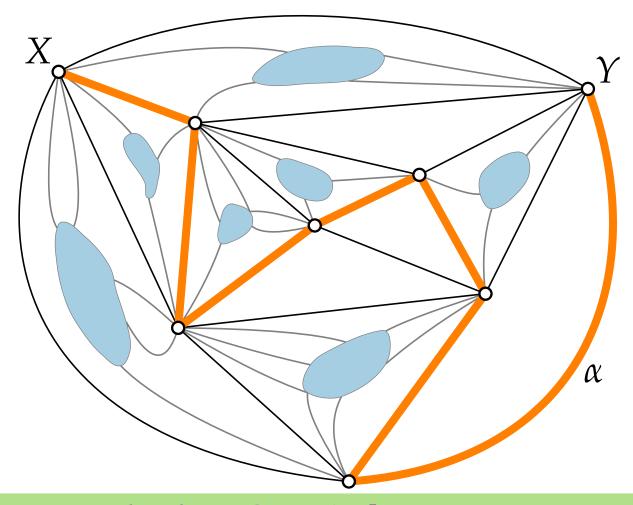


[Asano, Kikuchi & Saito '85]



[Asano, Kikuchi & Saito '85]

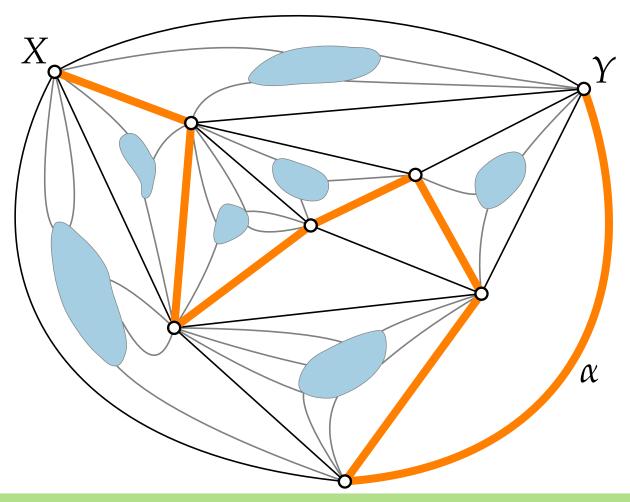
4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.



k vertices

[Asano, Kikuchi & Saito '85]

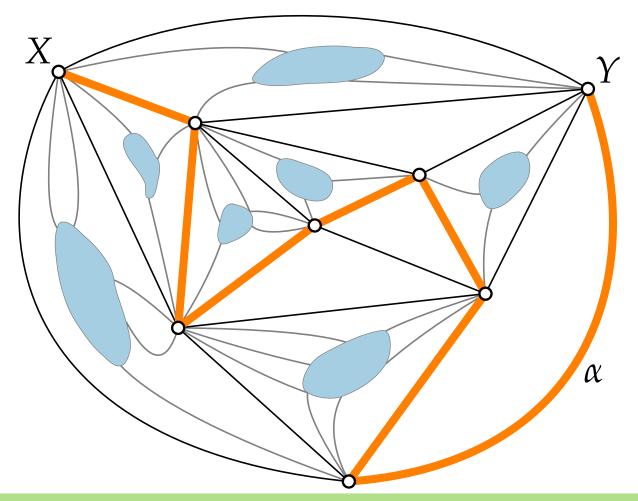
4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.



k vertices 2k - 5 int. faces

[Asano, Kikuchi & Saito '85]

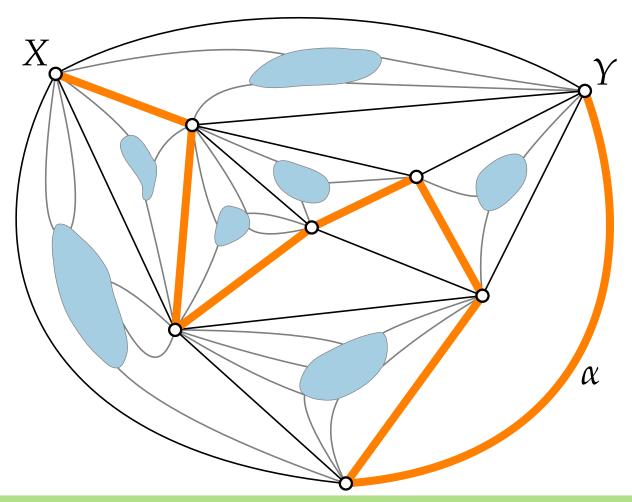
4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.



k vertices 2k - 5 int. faces k - 3 int. vtcs

[Asano, Kikuchi & Saito '85]

4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.



k vertices

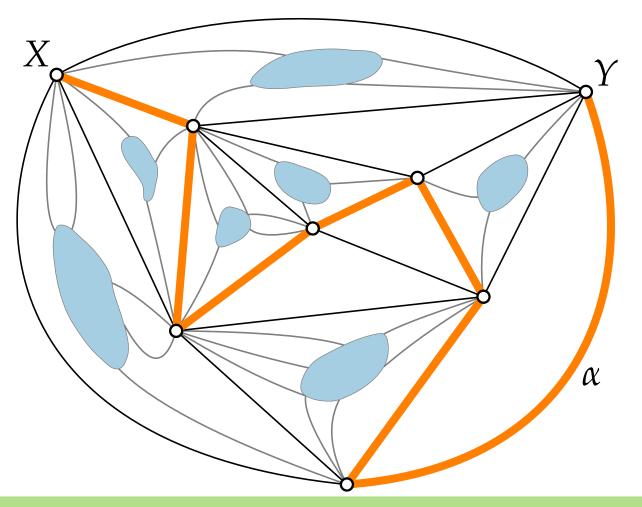
2k - 5 int. faces

k-3 int. vtcs

k-2 int. edges in P

[Asano, Kikuchi & Saito '85]

4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.



k vertices

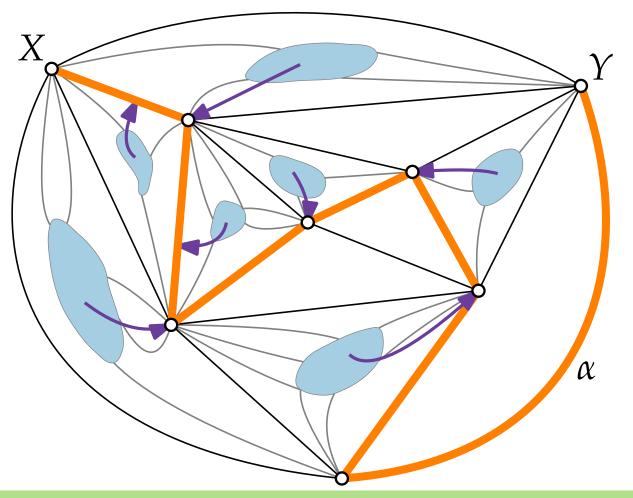
-2k - 5 int. faces

-k-3 int. vtcs

-k-2 int. edges in P

[Asano, Kikuchi & Saito '85]

4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.



k vertices

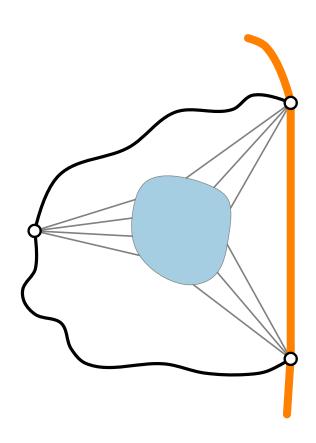
-2k - 5 int. faces

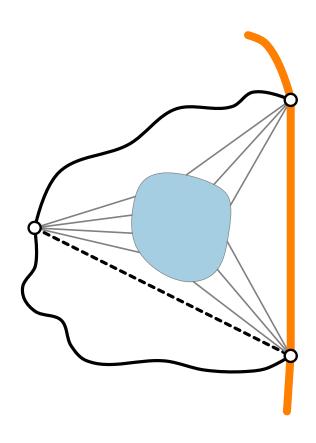
-k-3 int. vtcs

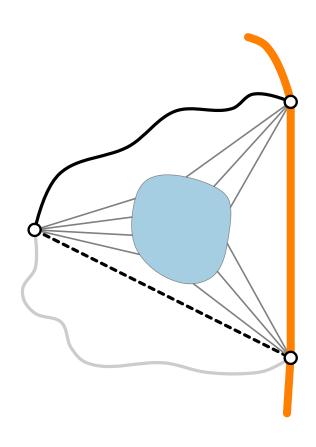
-k-2 int. edges in P

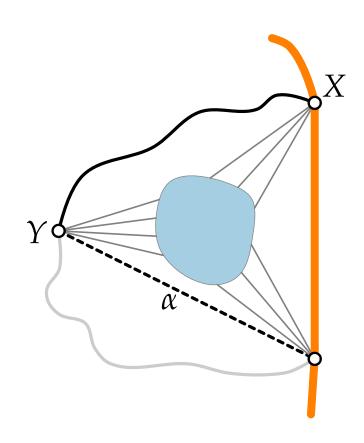
[Asano, Kikuchi & Saito '85]

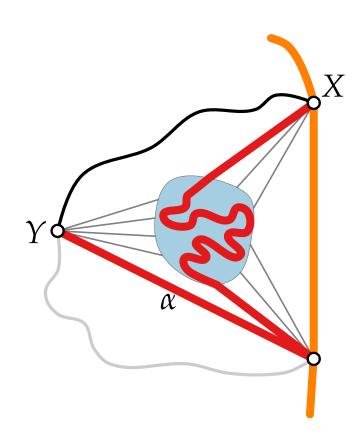
4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.

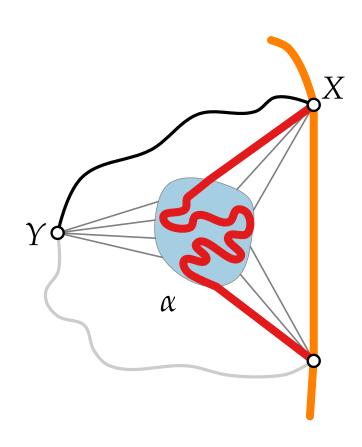


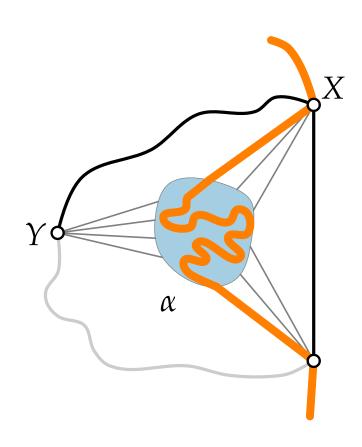


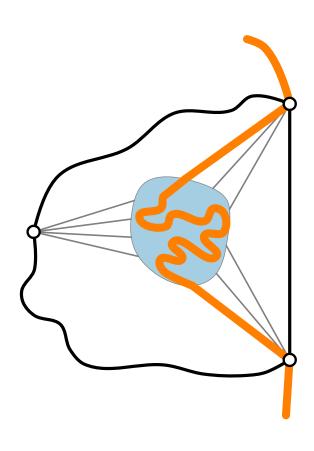


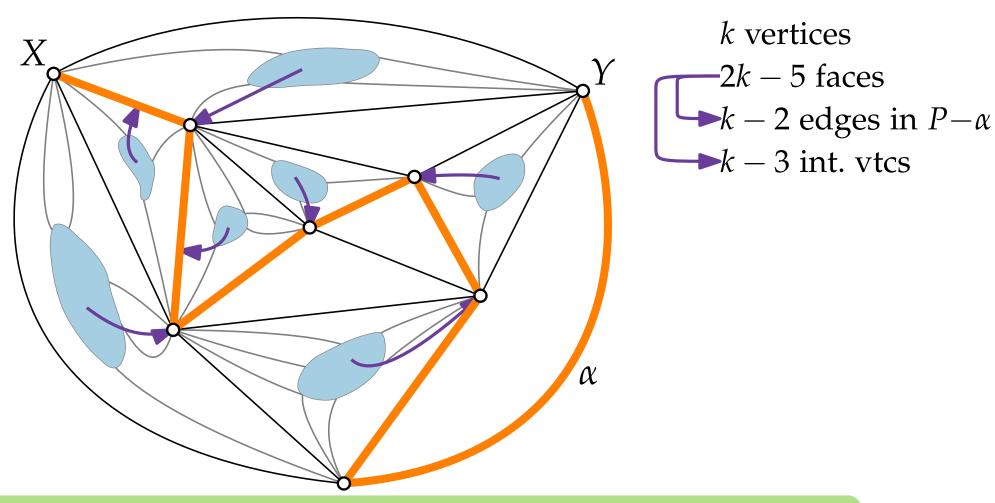






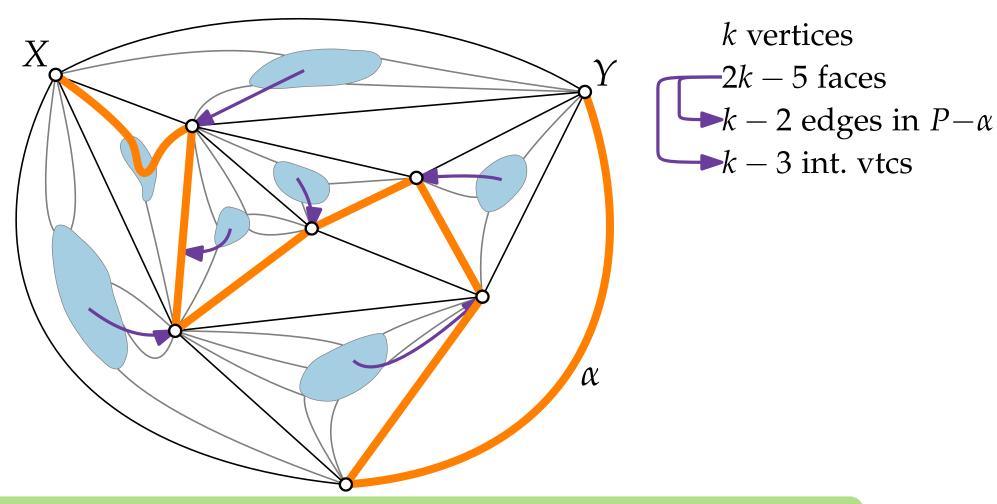






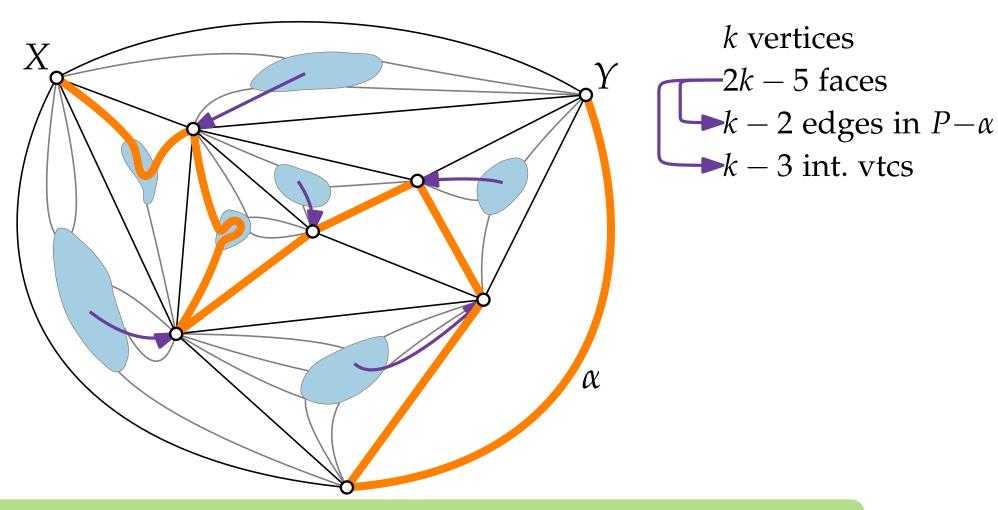
[Asano, Kikuchi & Saito '85]

4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.



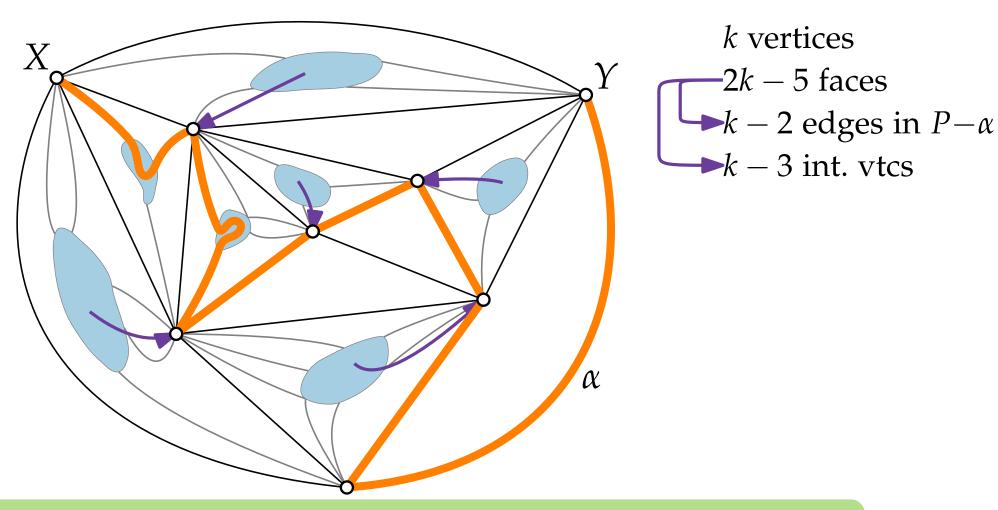
[Asano, Kikuchi & Saito '85]

4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.



[Asano, Kikuchi & Saito '85]

4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.

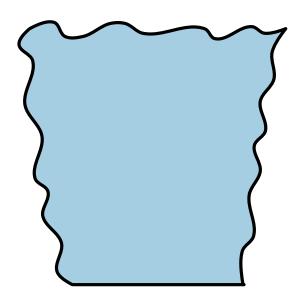


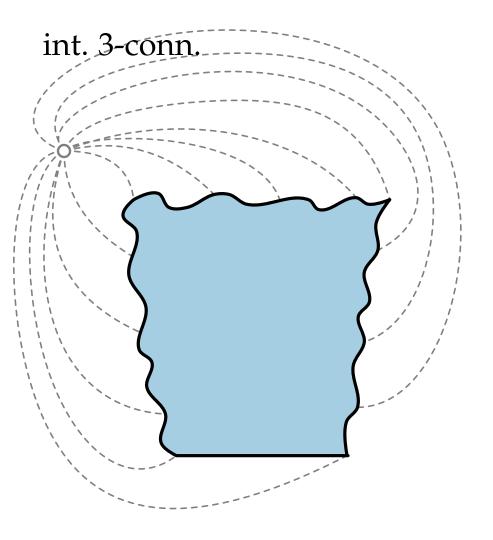
[Asano, Kikuchi & Saito '85]

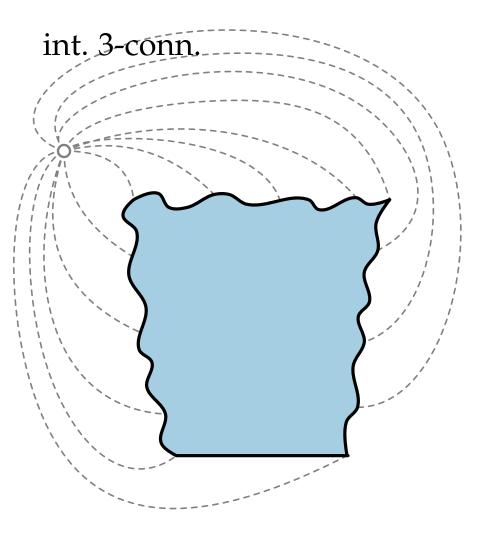
4-conn. triangulation \Rightarrow Hamiltonian path in O(n) time.

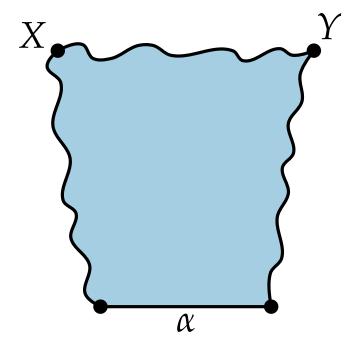
triangulation $\Rightarrow T_{\text{int}}$ -path in O(n) time.

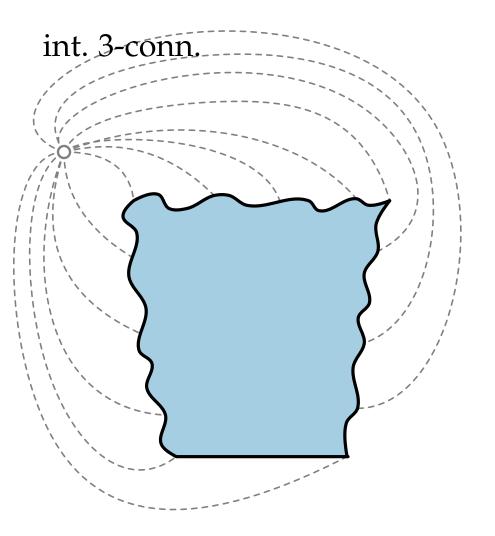
int. 3-conn.

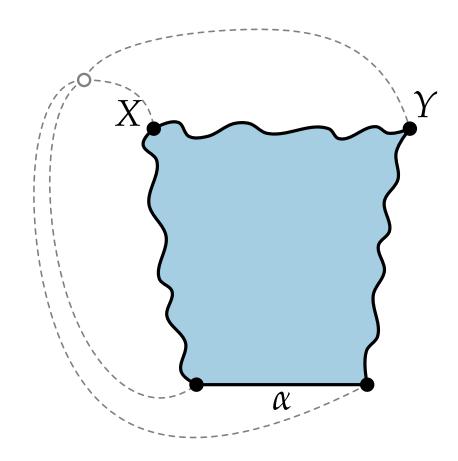


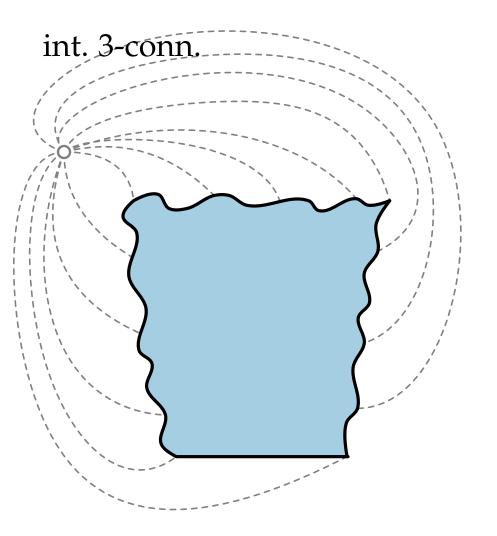


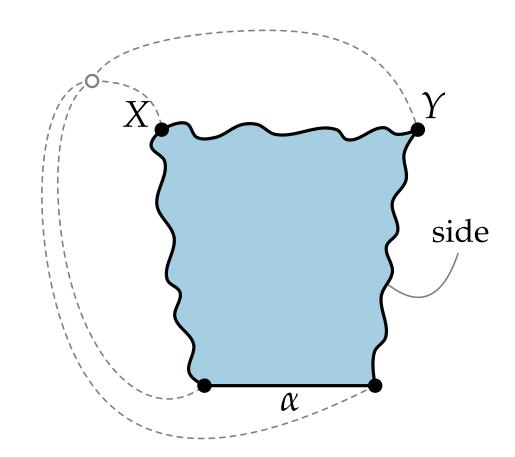


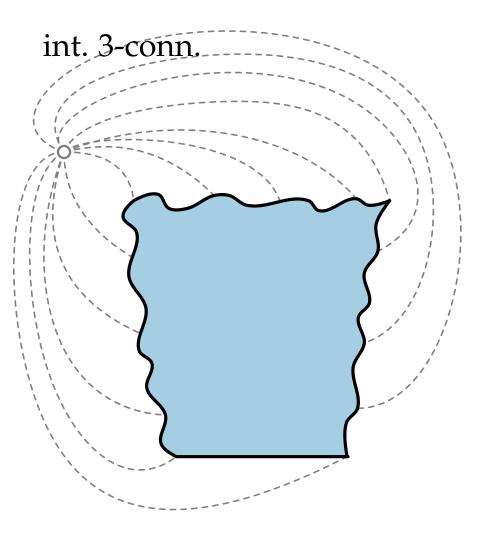


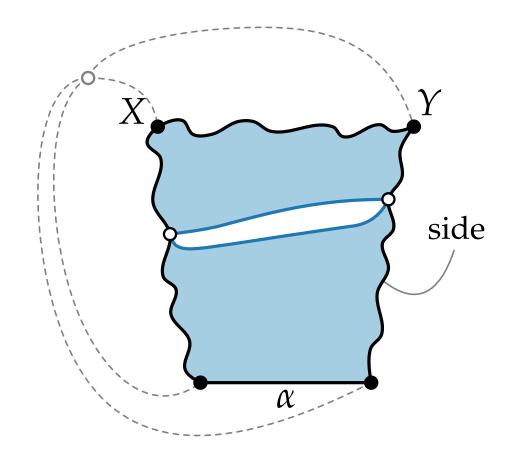


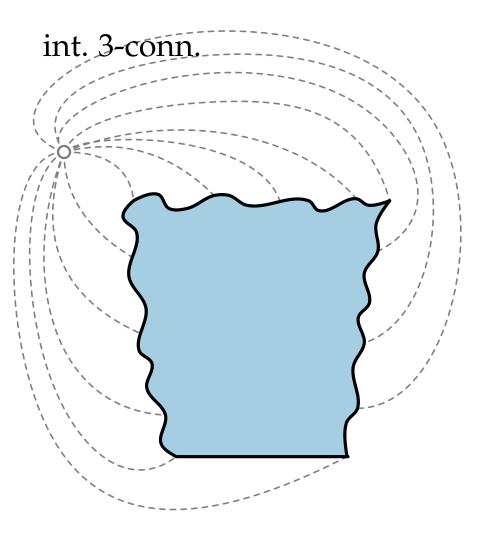


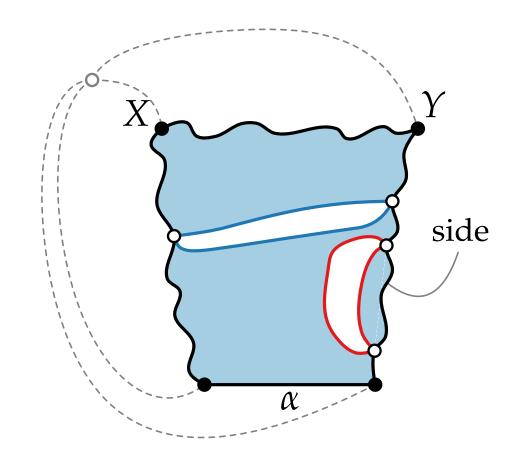


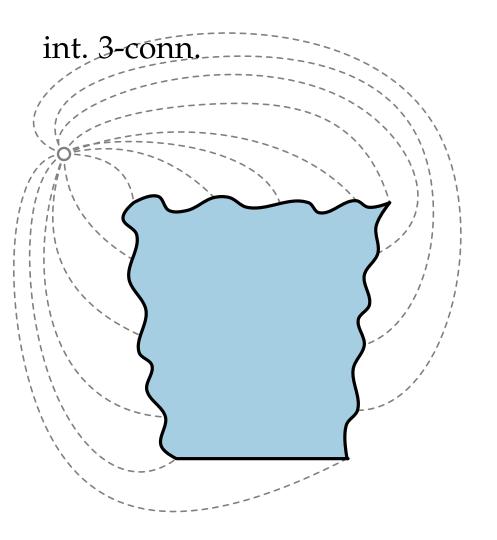


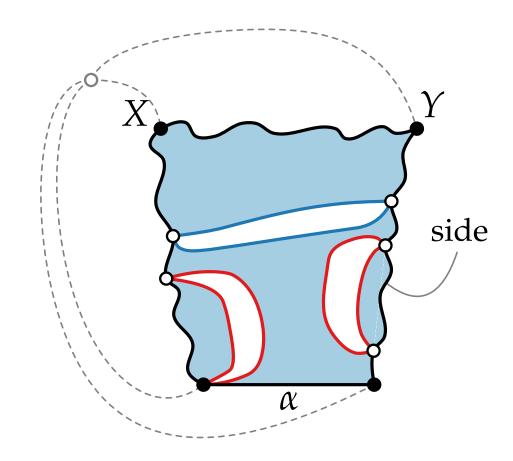


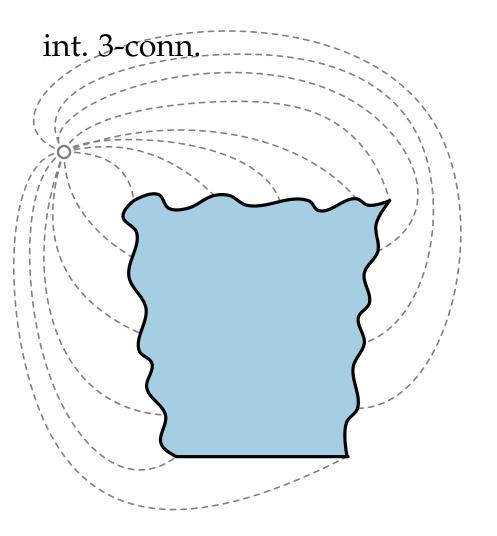


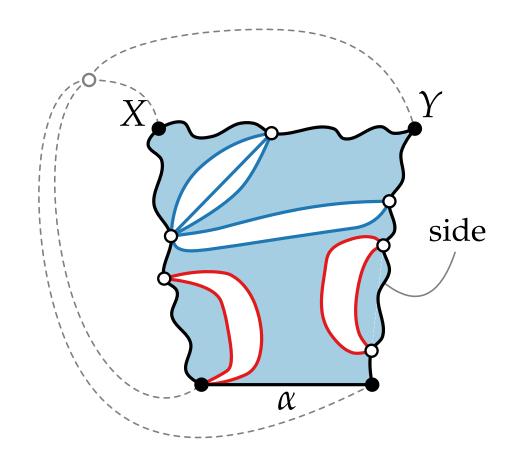


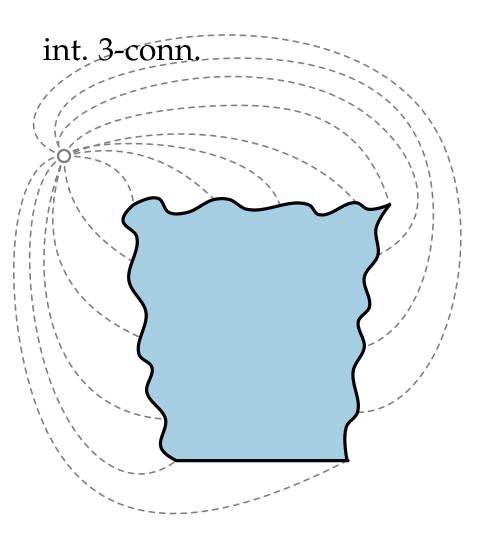


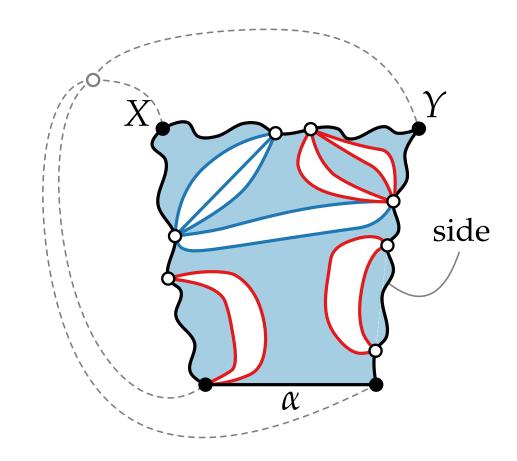


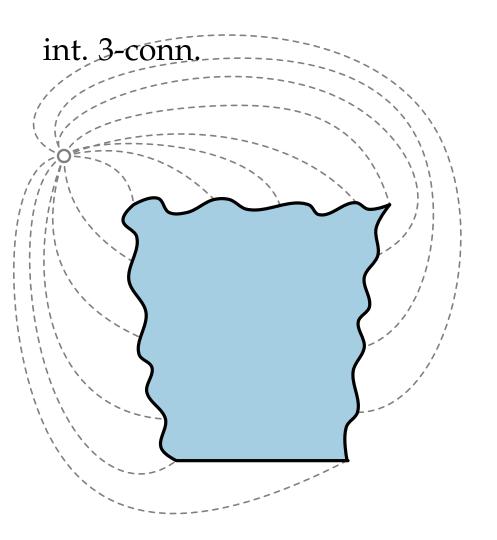


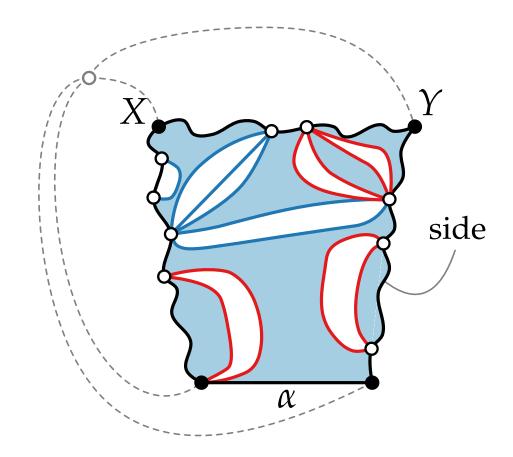


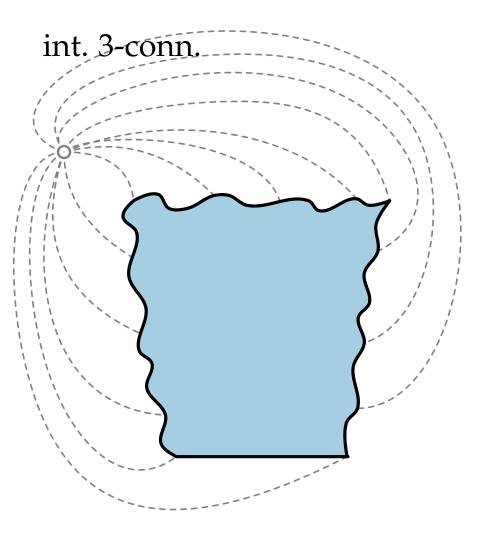


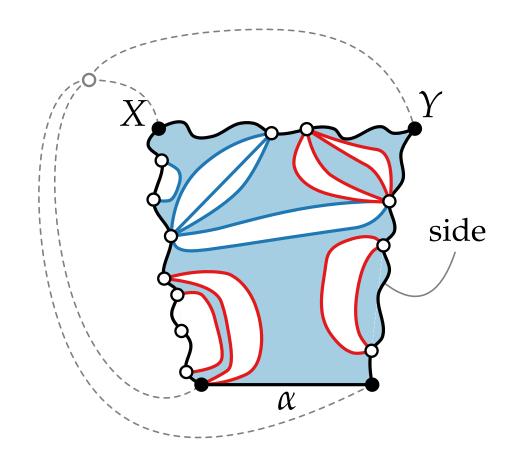


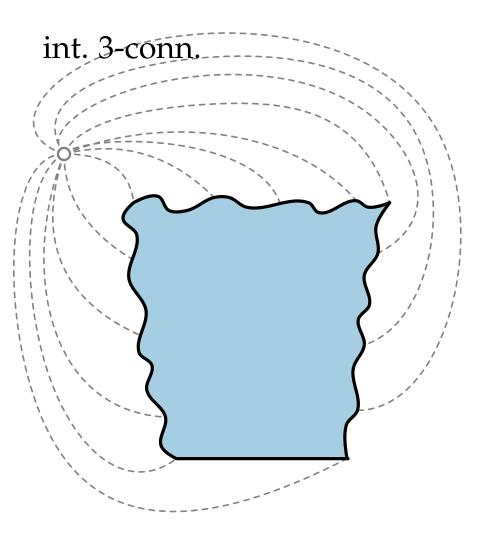




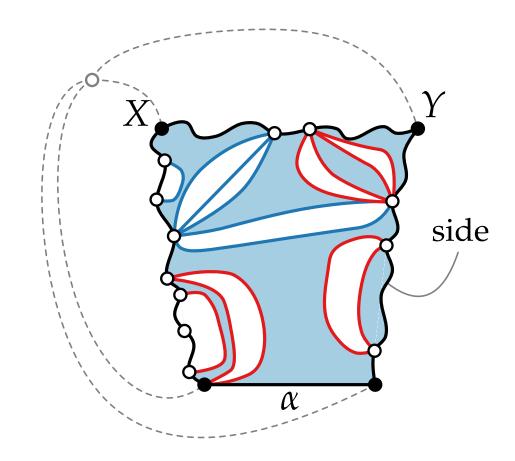




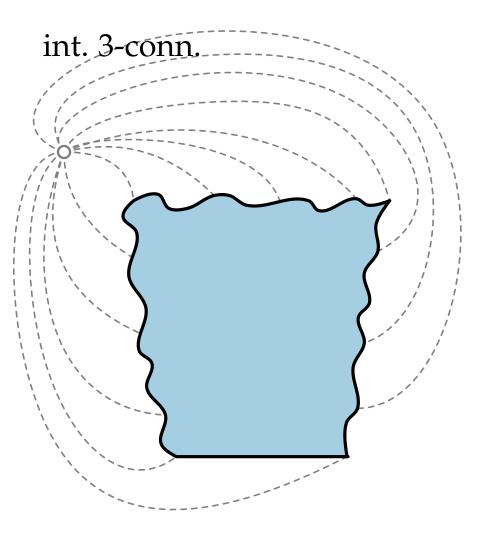




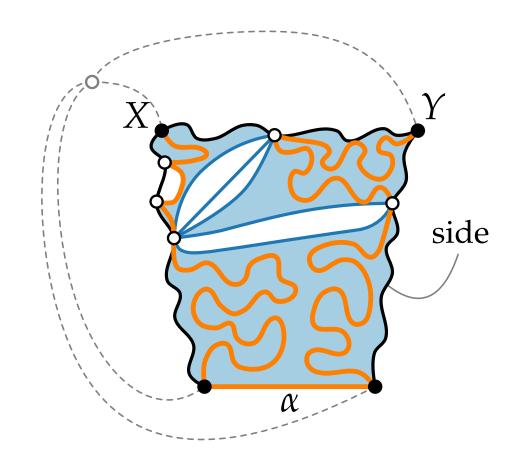
corner-3-conn.



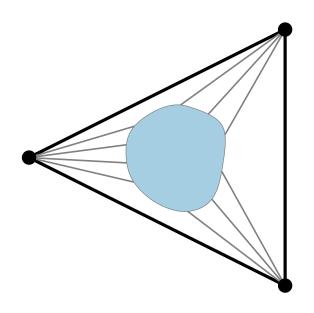
G is corner-3-conn., X, Y, α on outer face \Rightarrow T_{int} -path

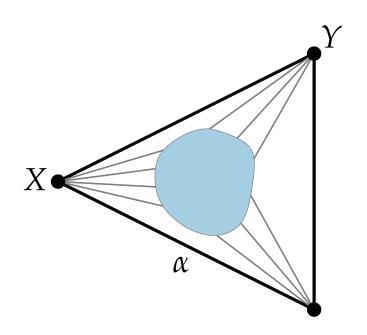


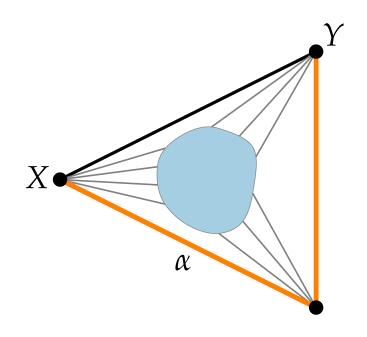
corner-3-conn.

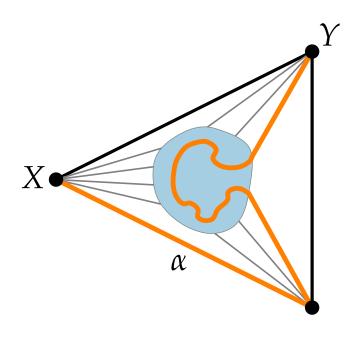


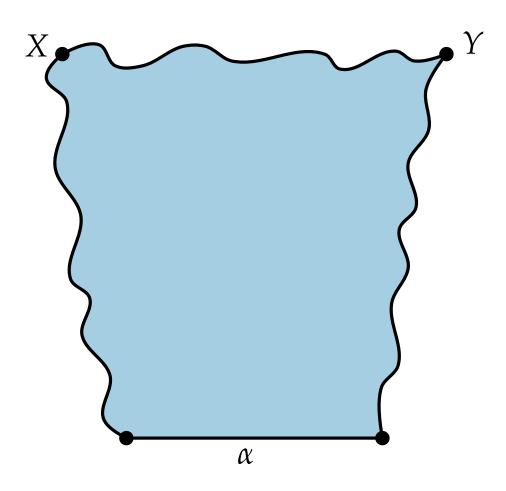
G is corner-3-conn., X, Y, α on outer face \Rightarrow T_{int} -path

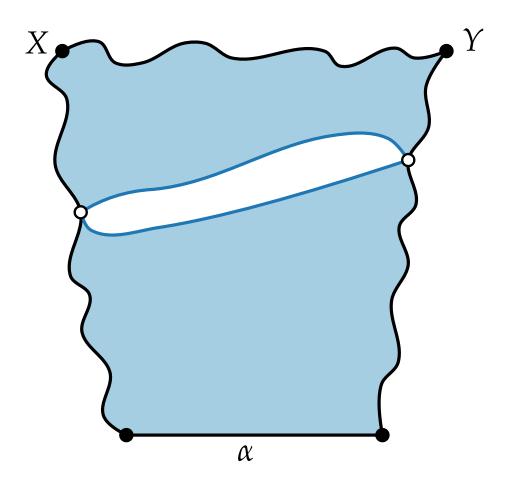


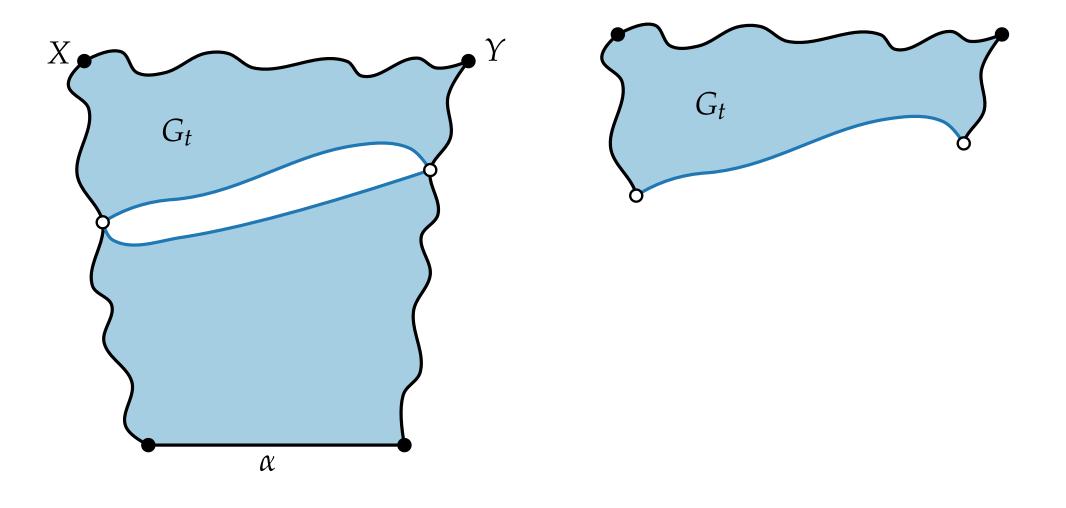


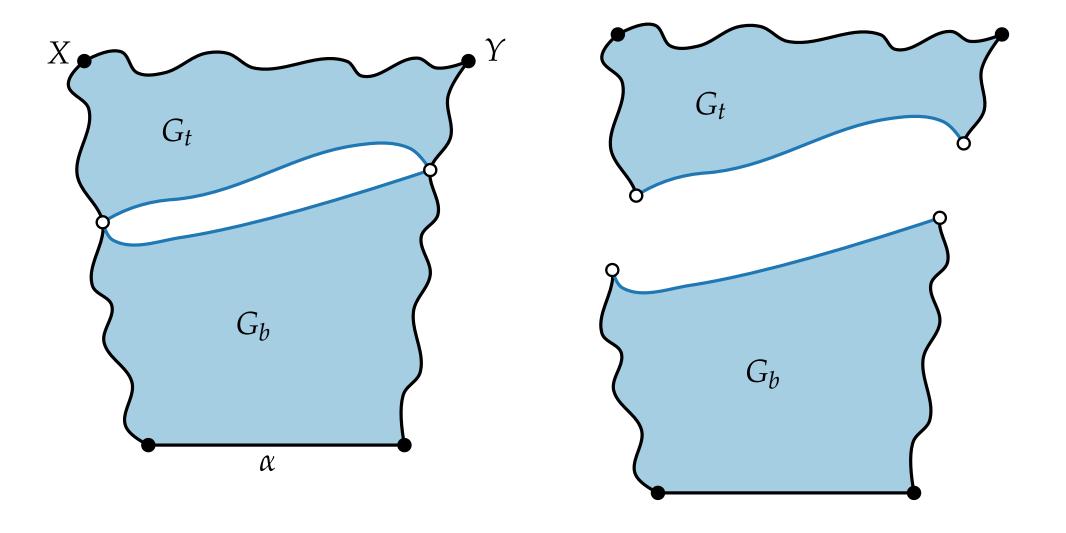


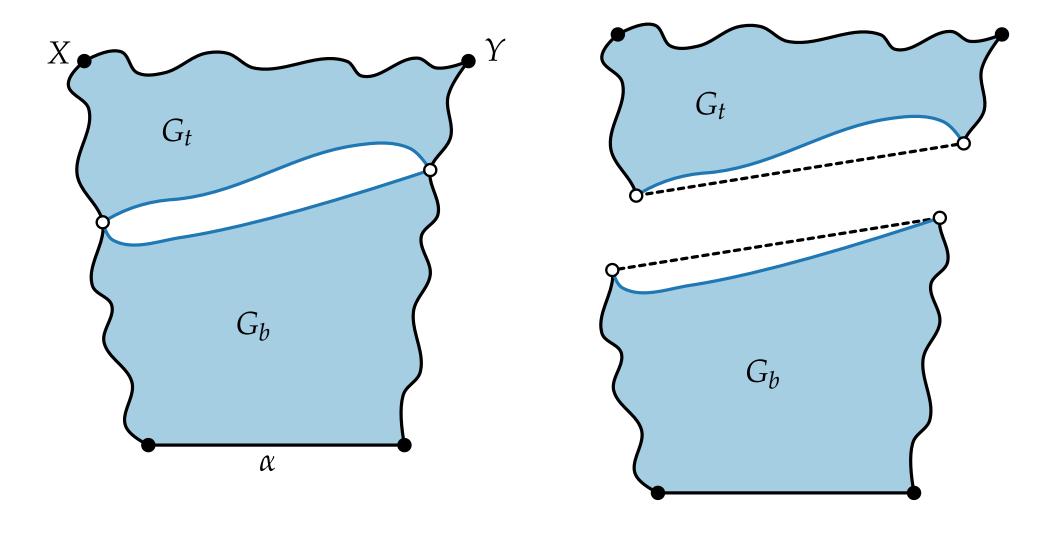


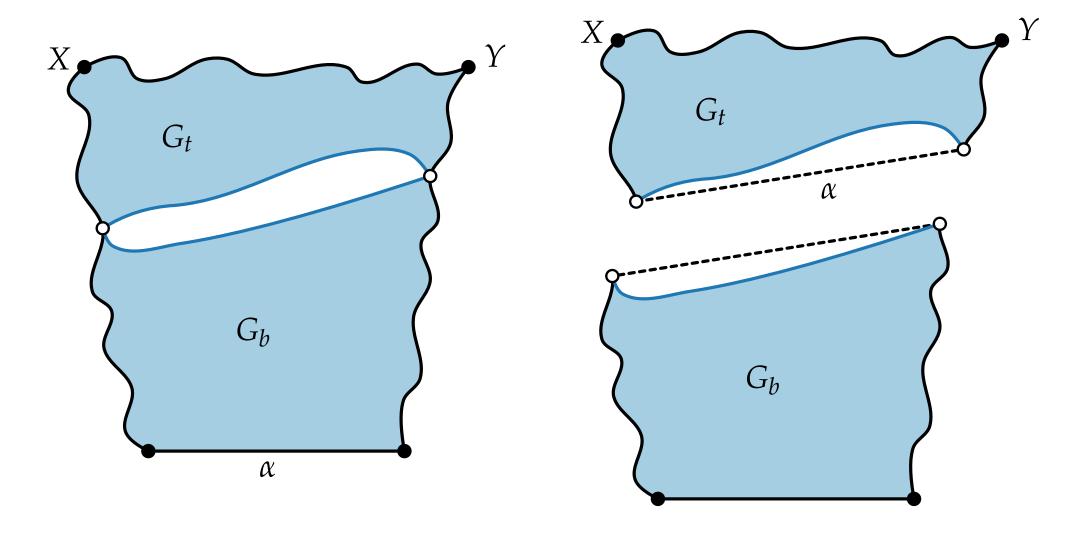


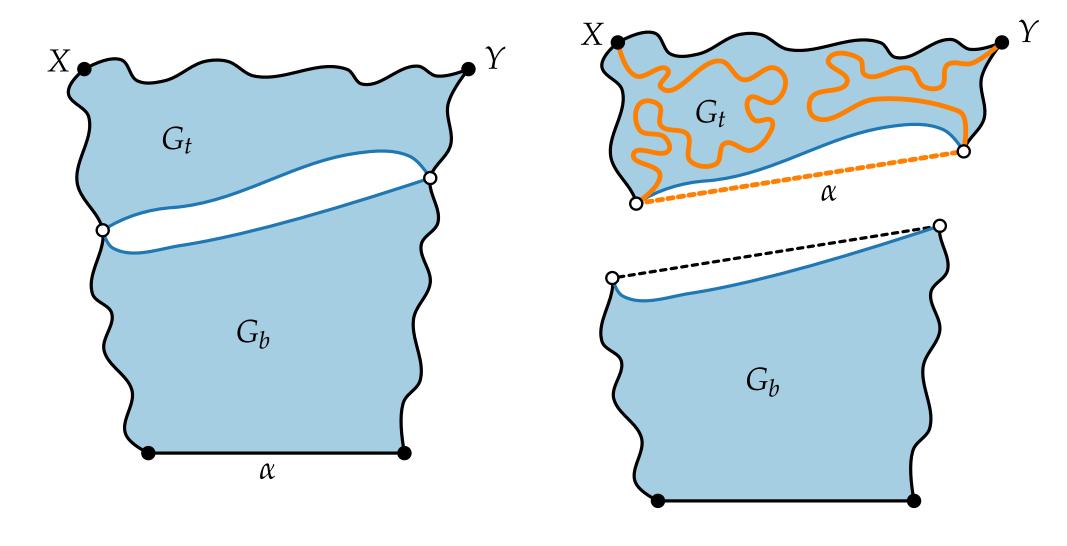


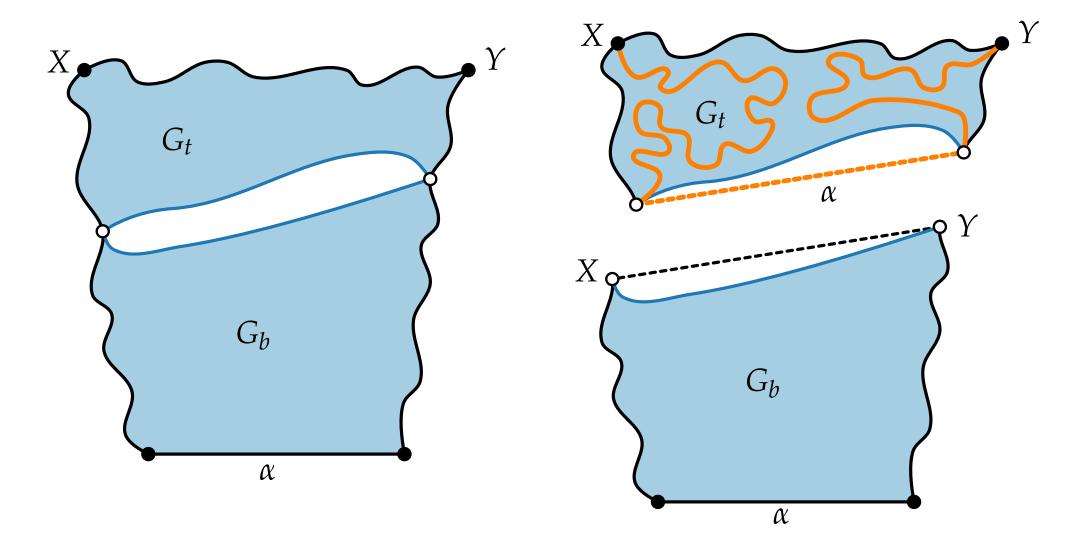


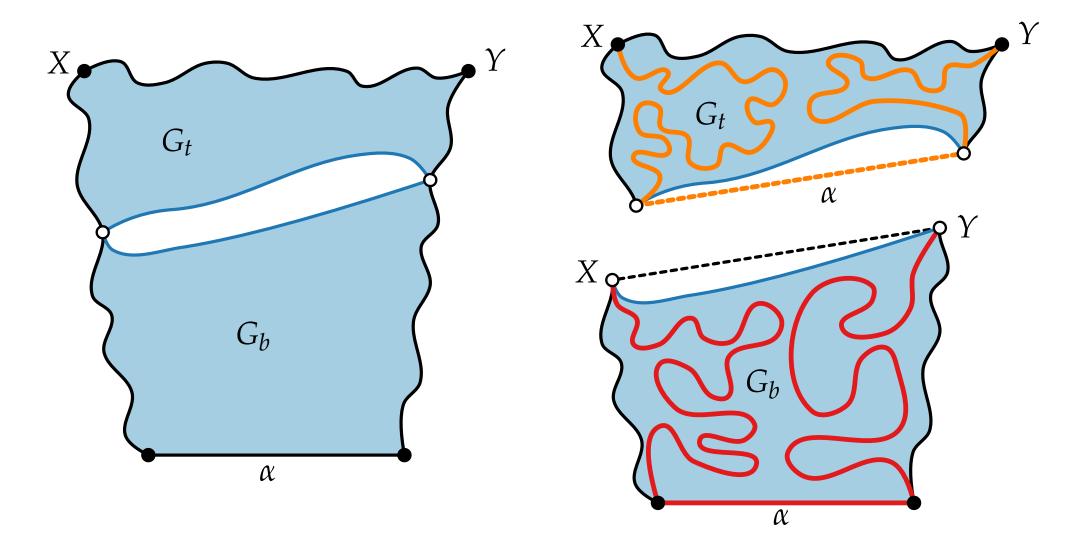


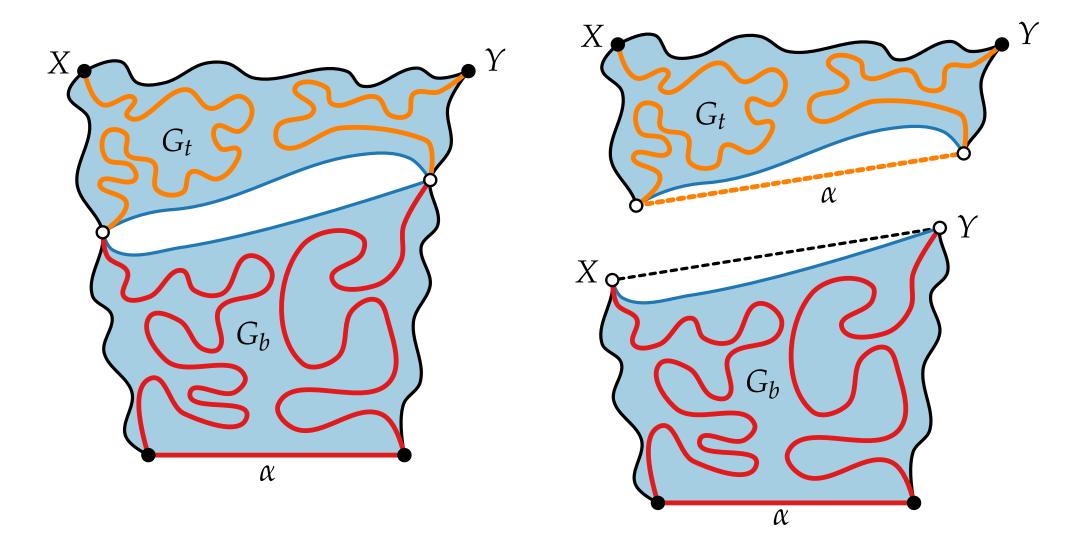


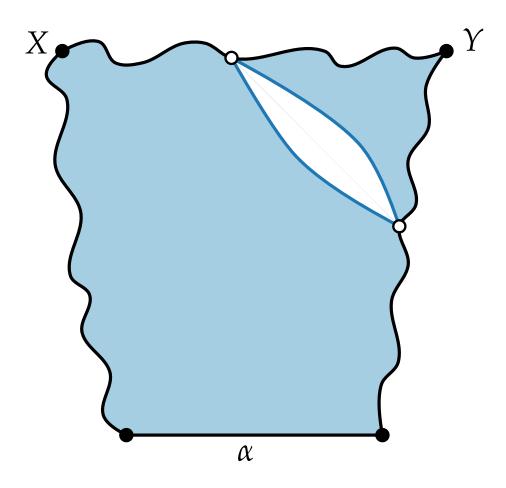


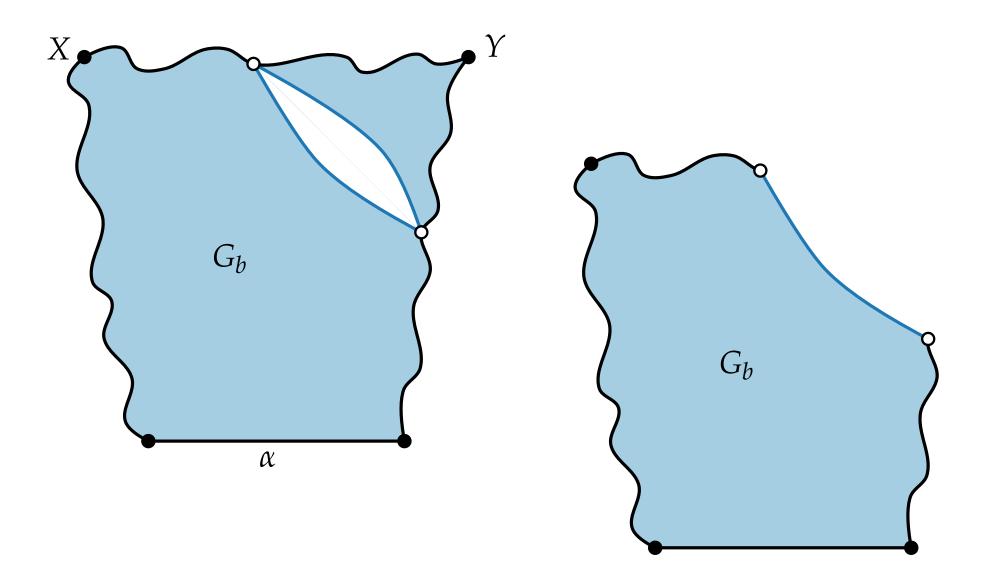


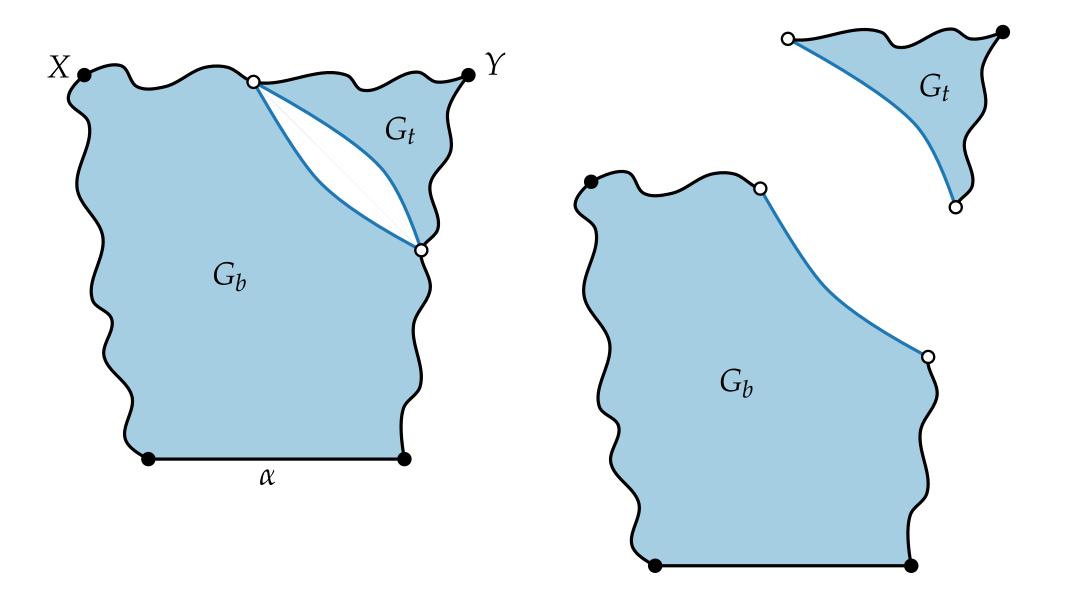


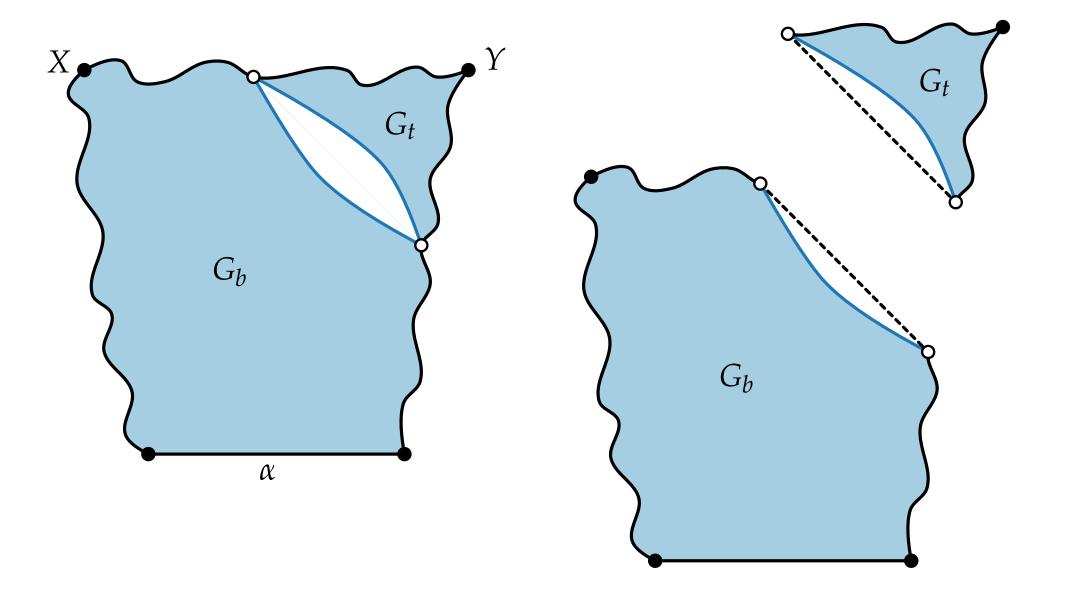


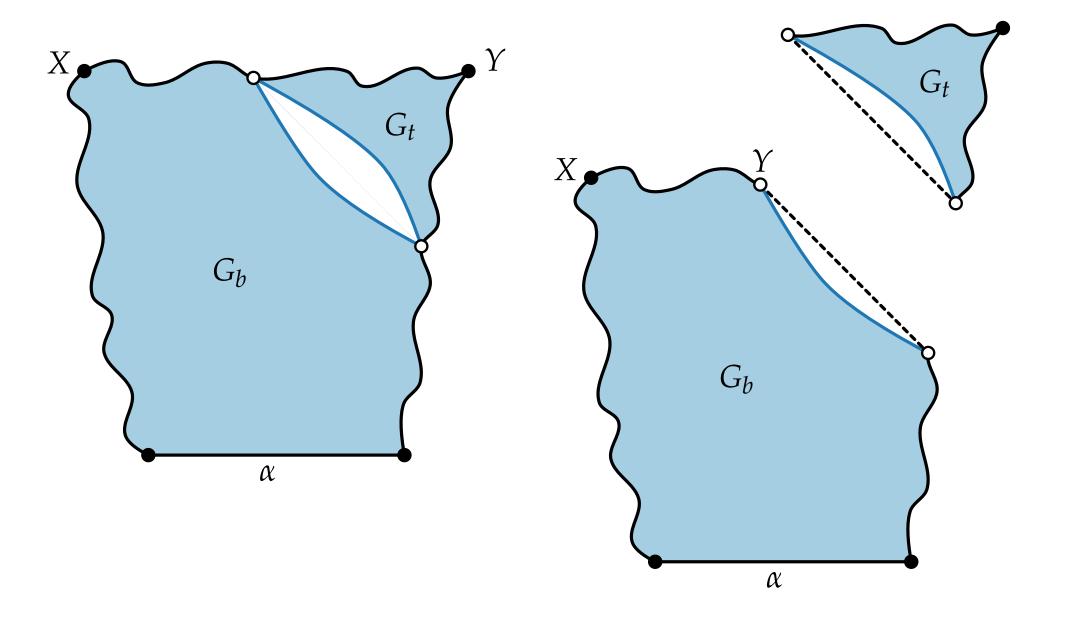


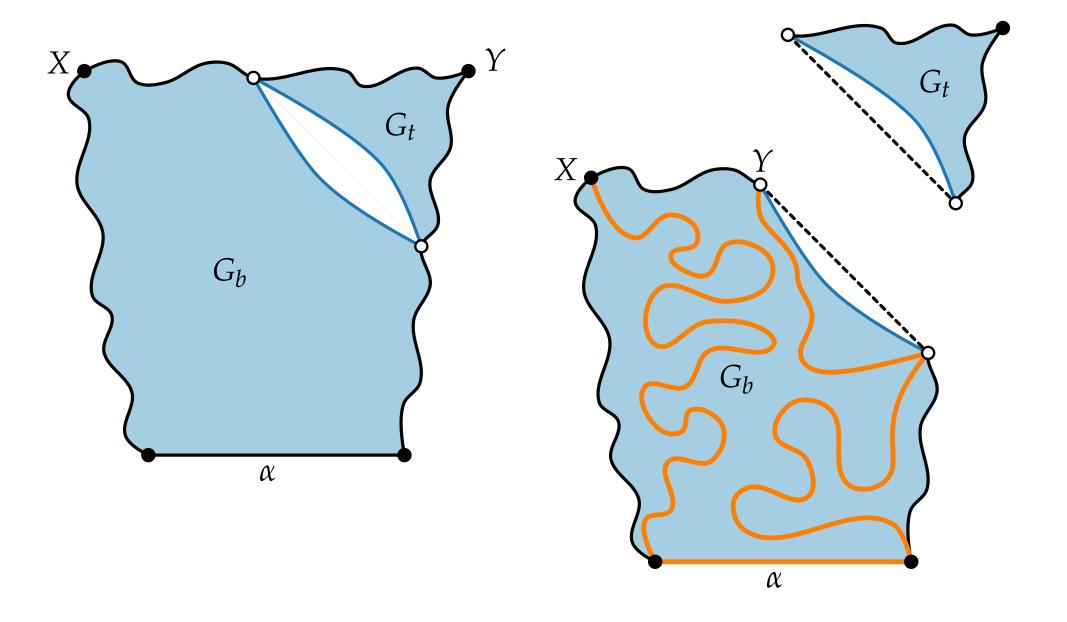


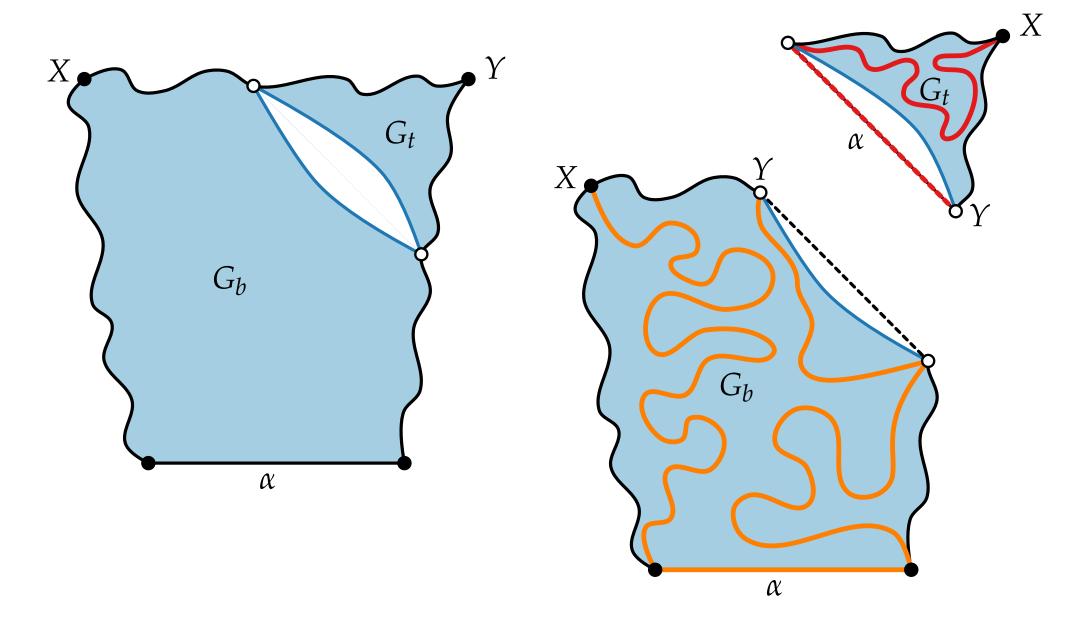


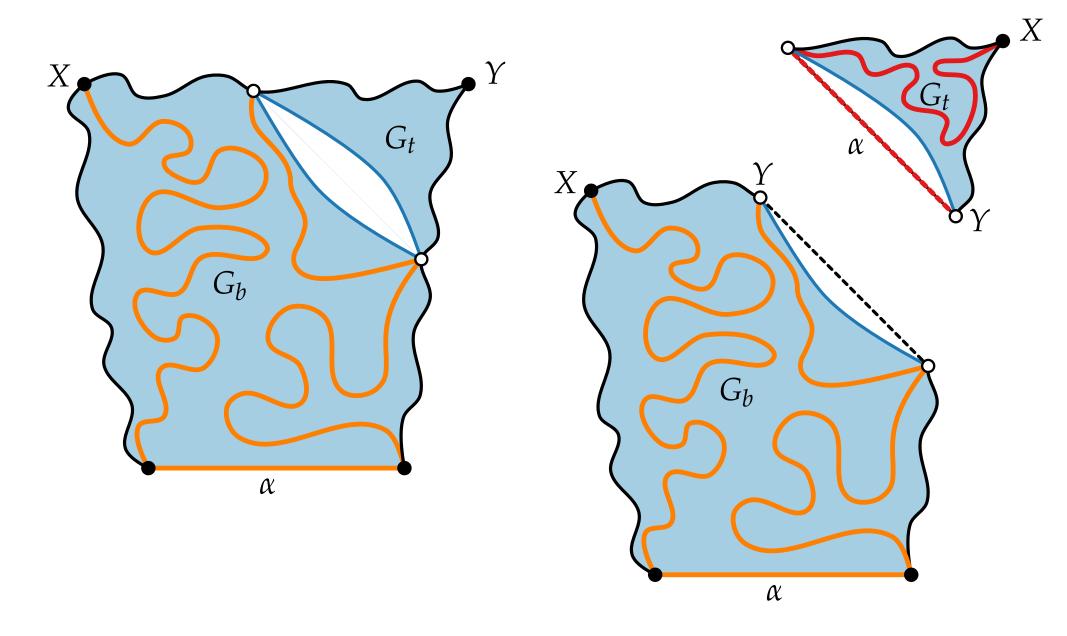


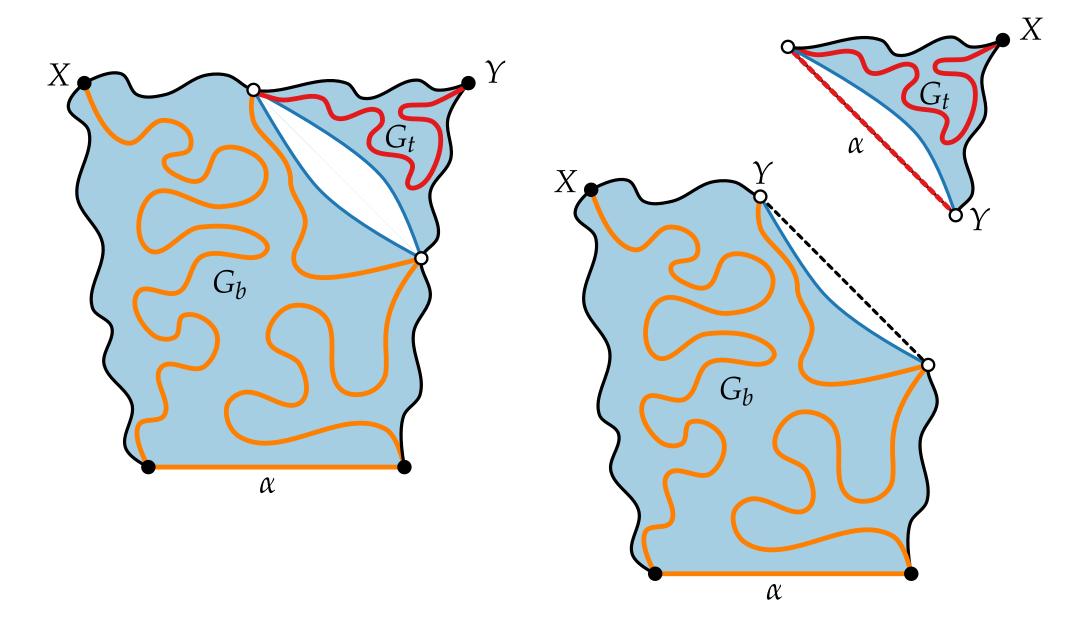


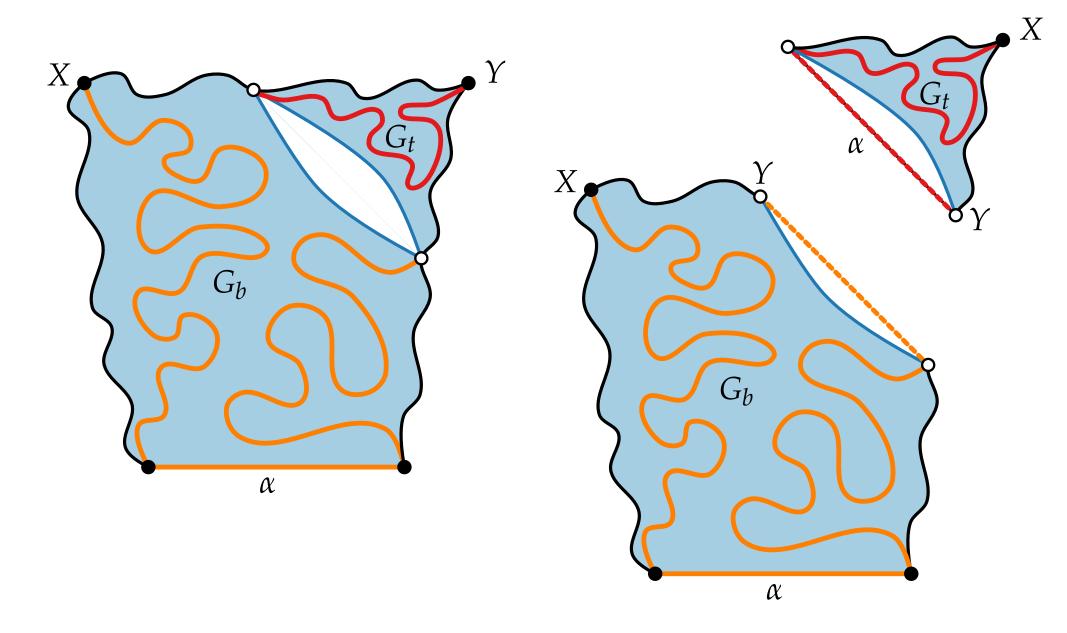


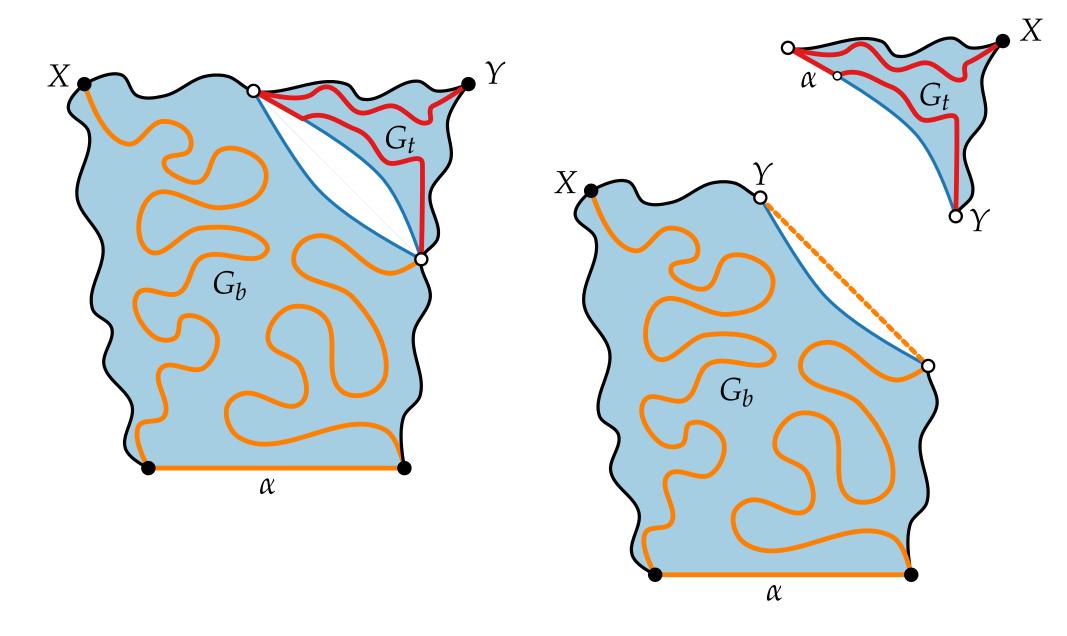




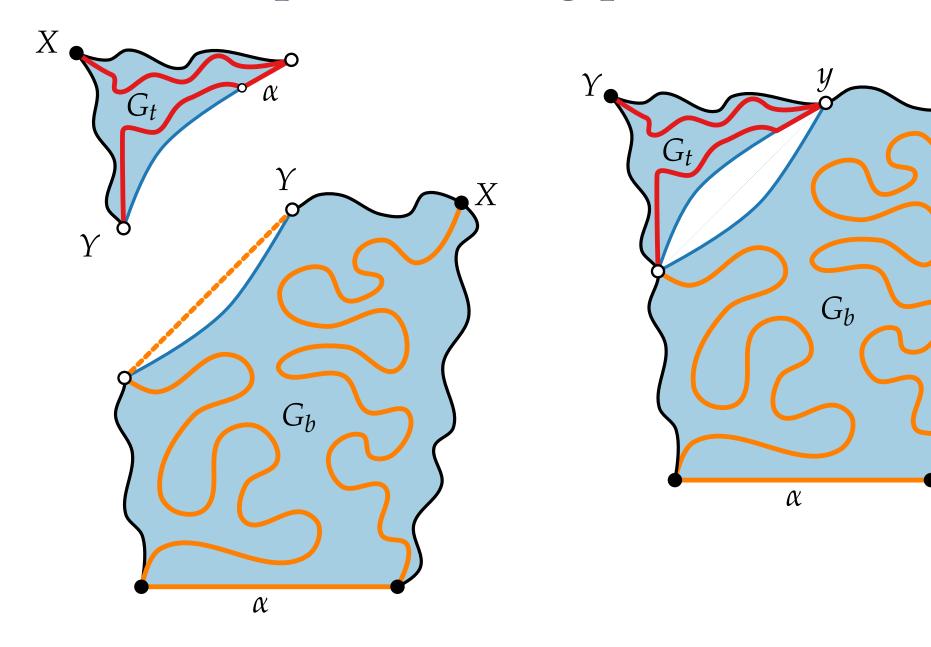




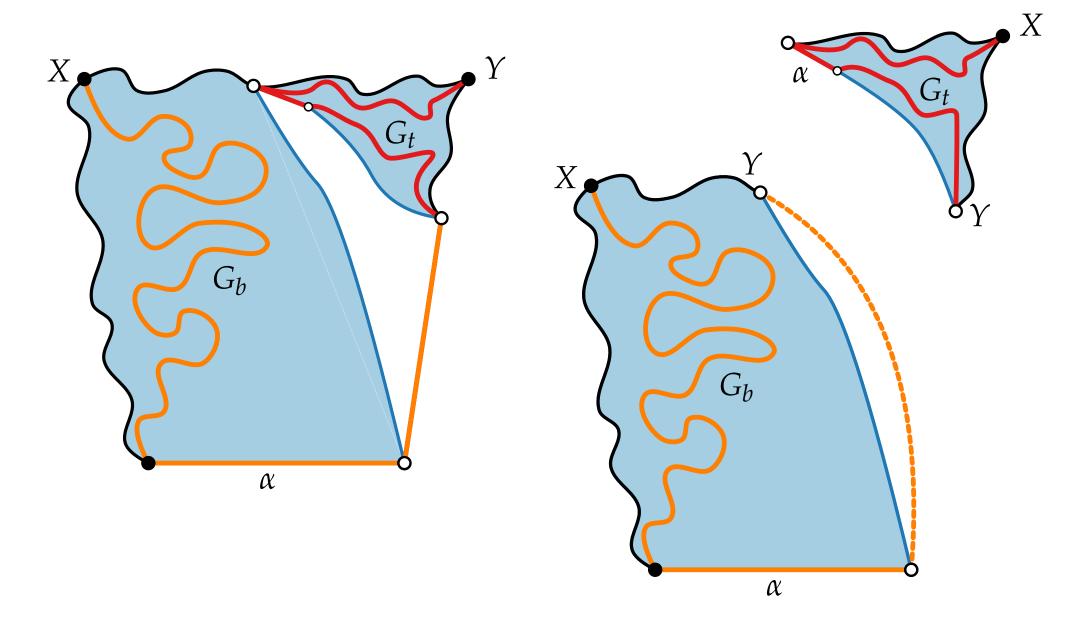




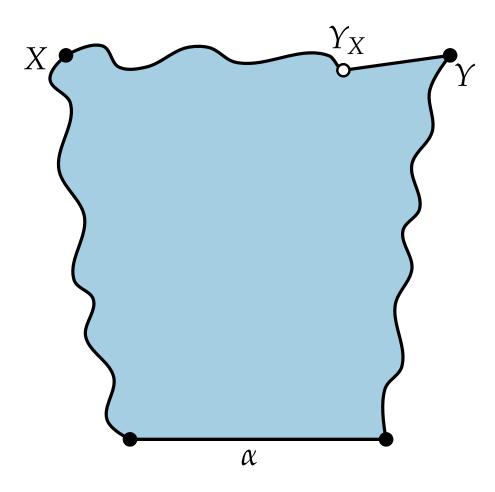
Case 3': top-left cutting pair



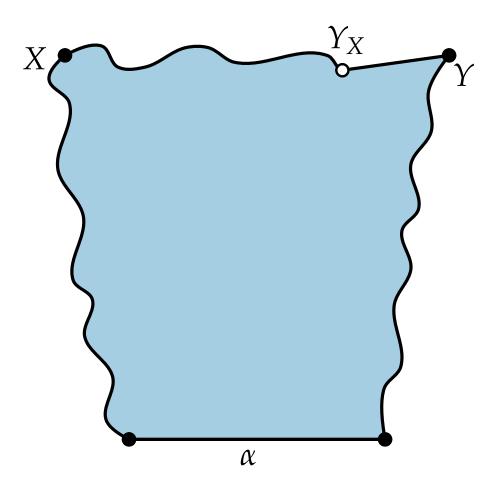
Case 3": top-bottom cutting pair



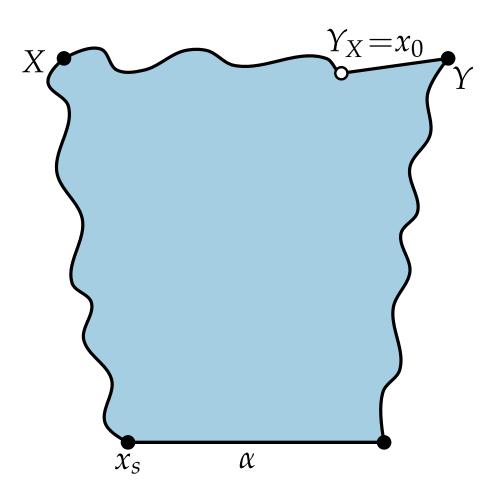
Case 4: No cutting pair



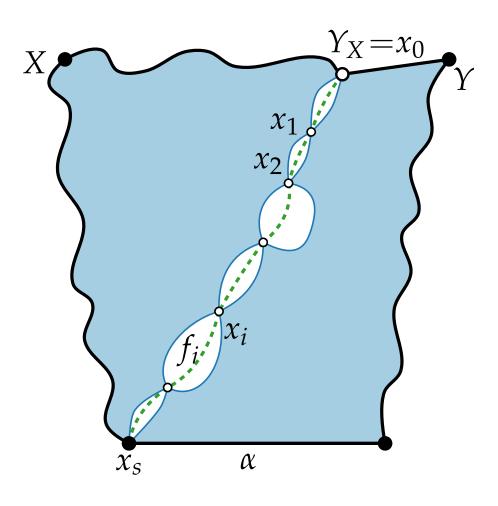
Necklace $\langle Y_X = x_0, f_1, x_1, \dots, x_{s-1}, f_s, x_s \rangle$

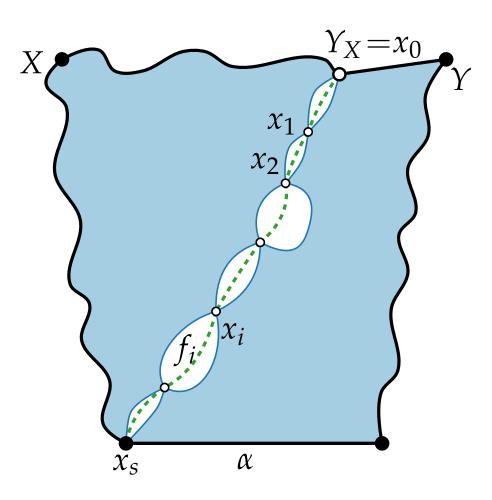


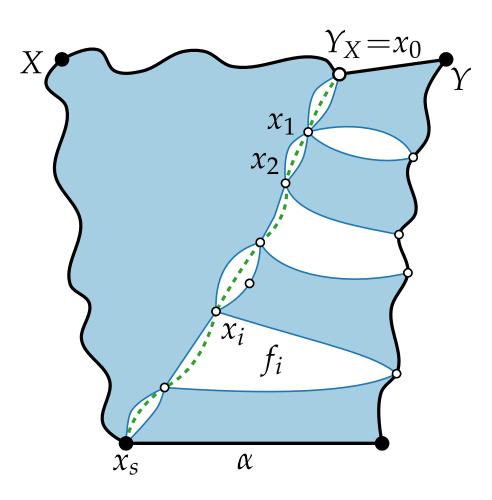
Necklace $\langle Y_X = x_0, f_1, x_1, \dots, x_{s-1}, f_s, x_s \rangle$

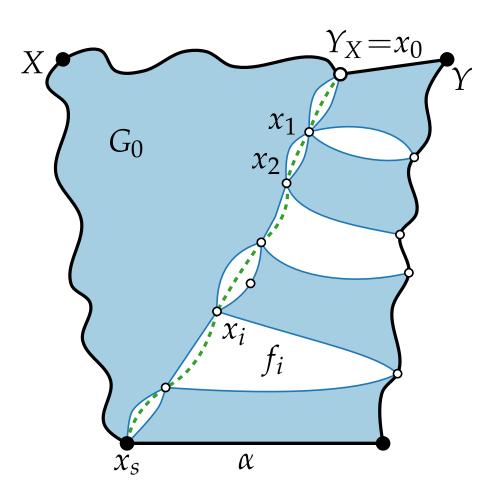


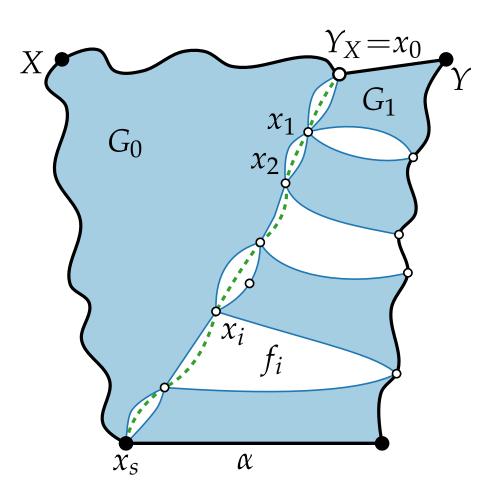
Necklace $\langle Y_X = x_0, f_1, x_1, \dots, x_{s-1}, f_s, x_s \rangle$

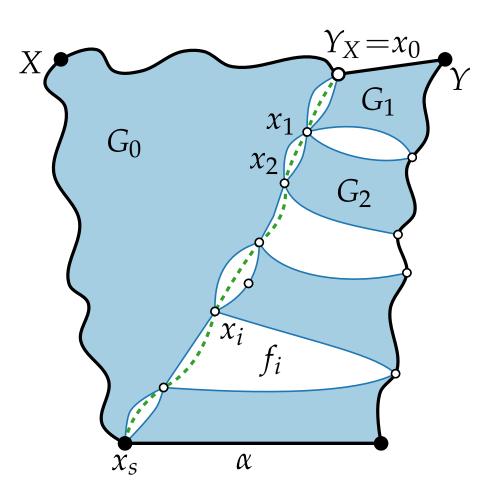


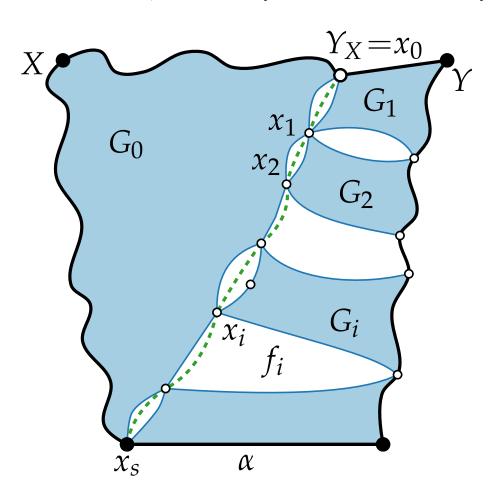


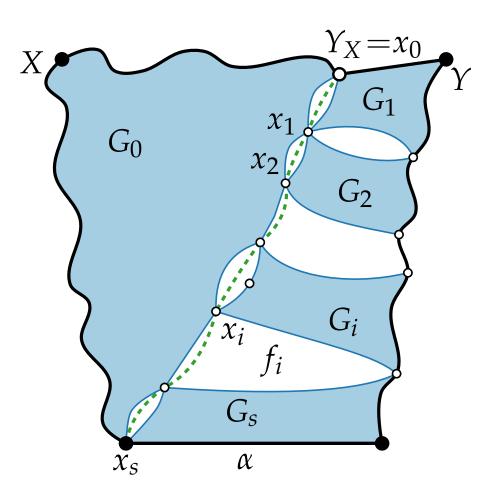


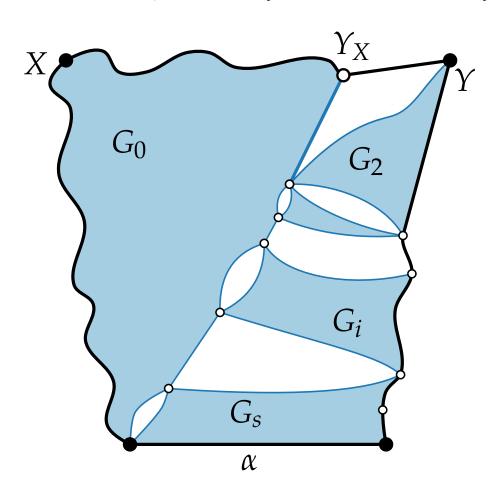


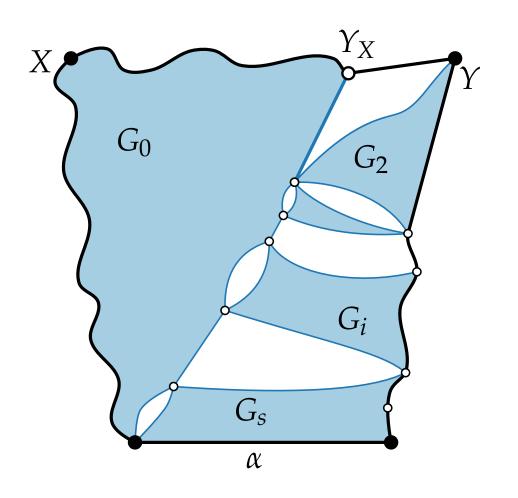


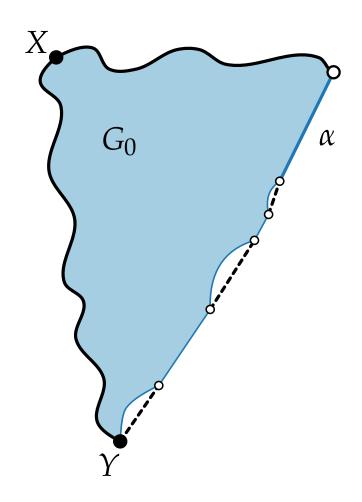


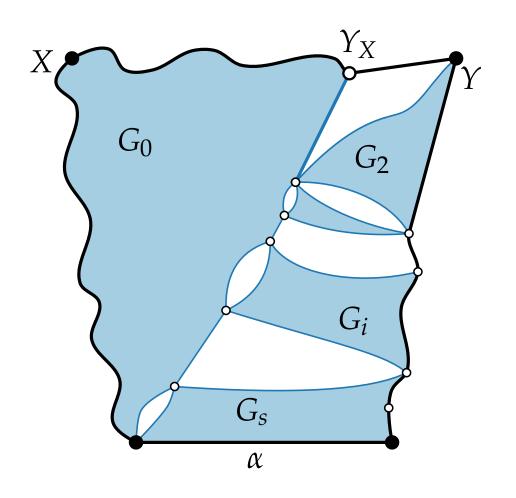


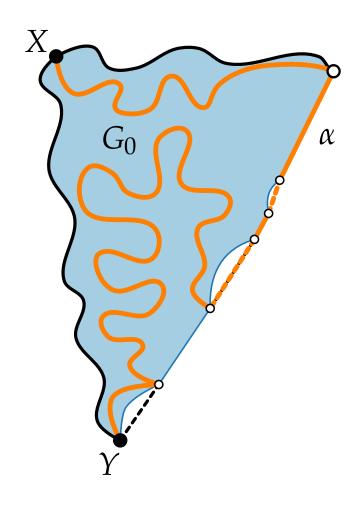


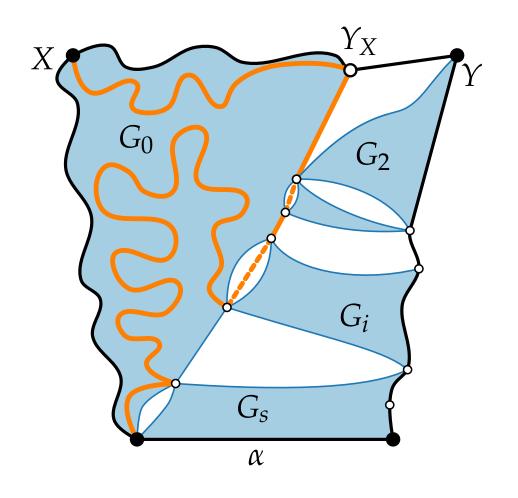


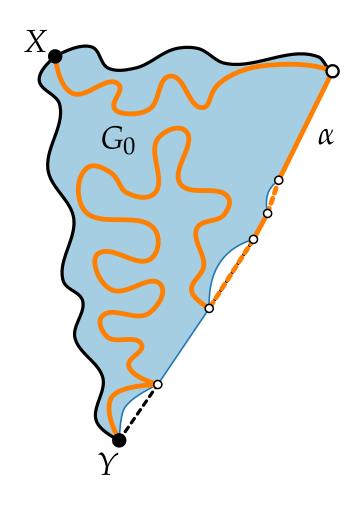


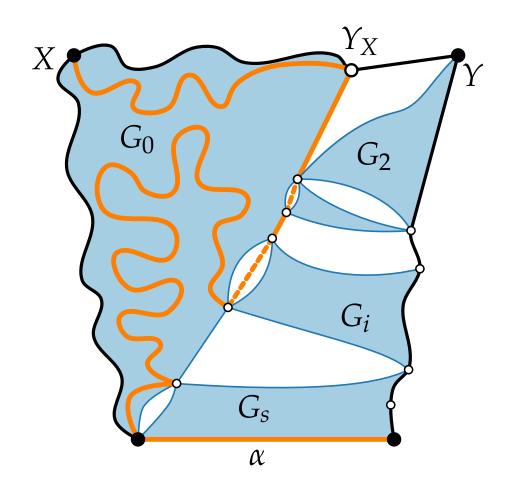


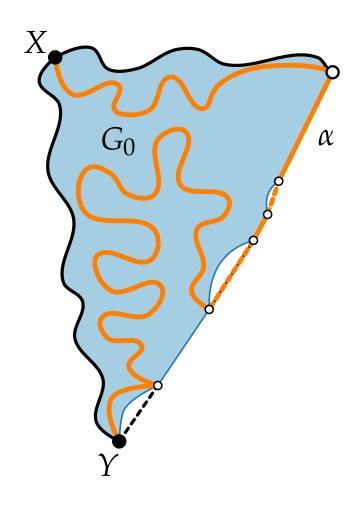


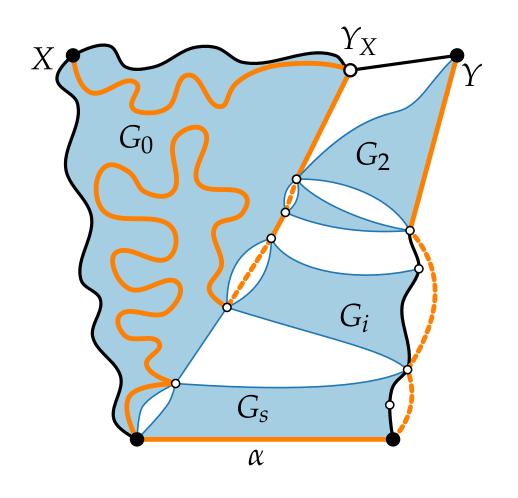


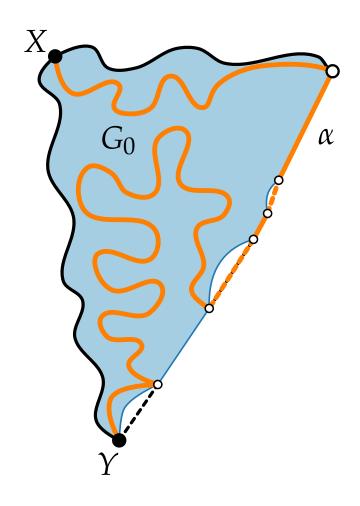


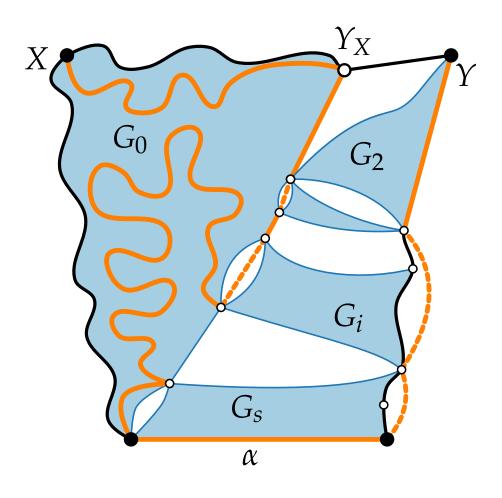


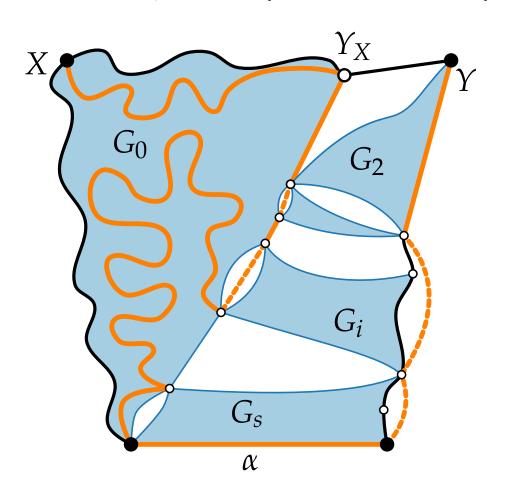




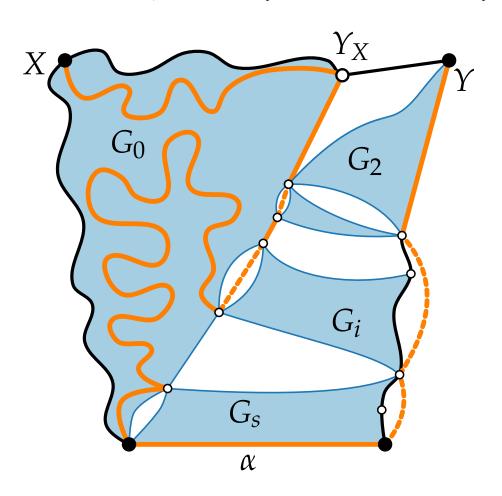


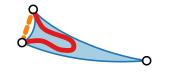


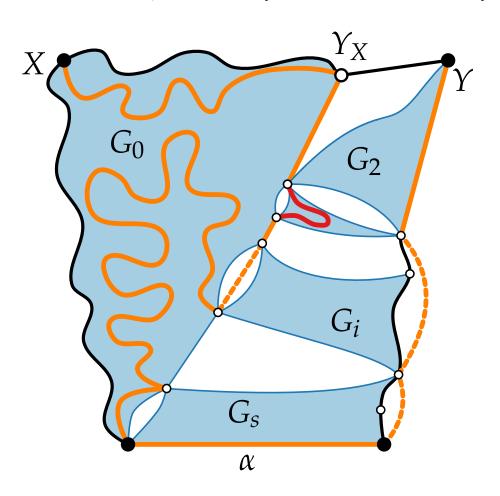




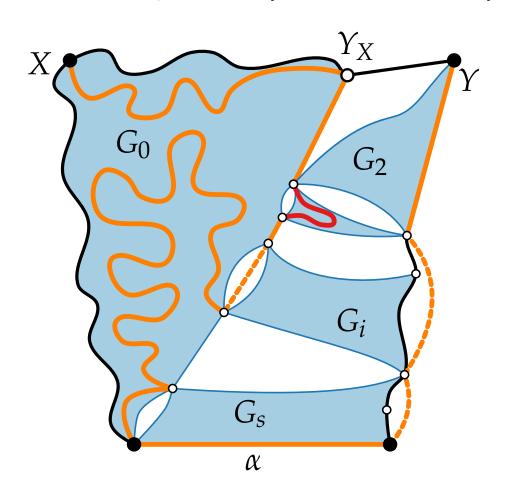


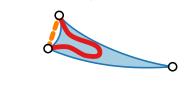


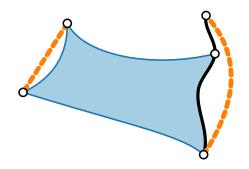


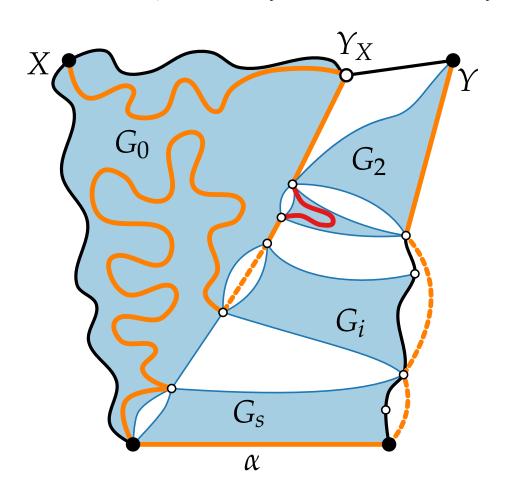




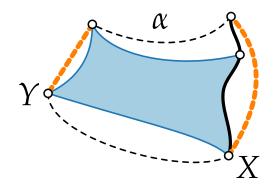


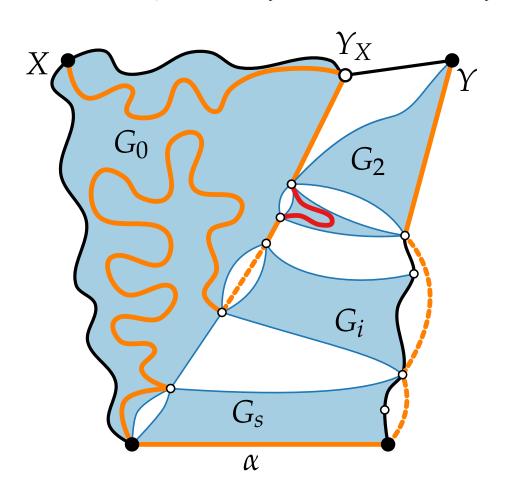




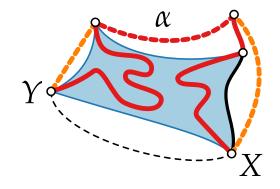


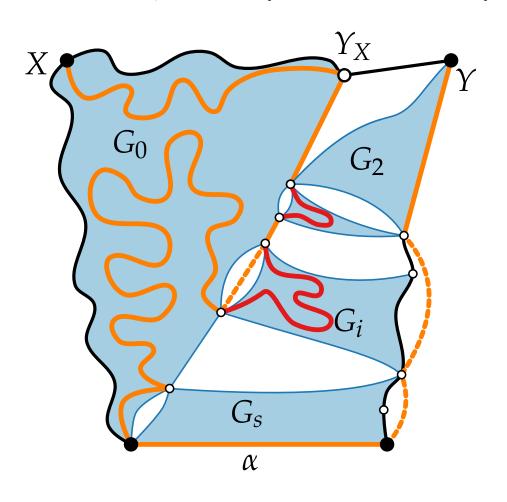




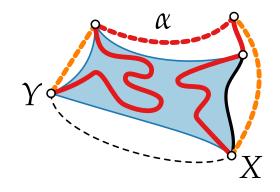


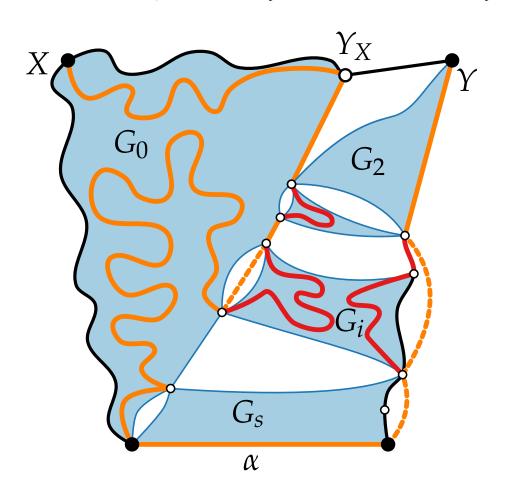




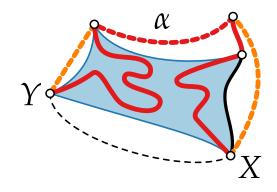


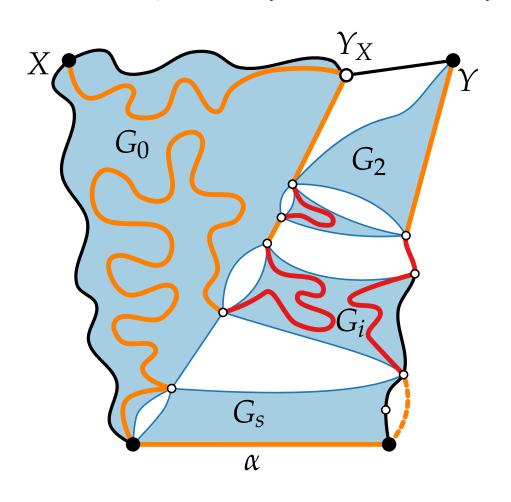




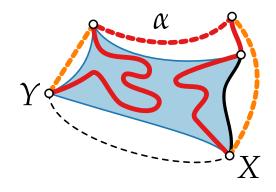


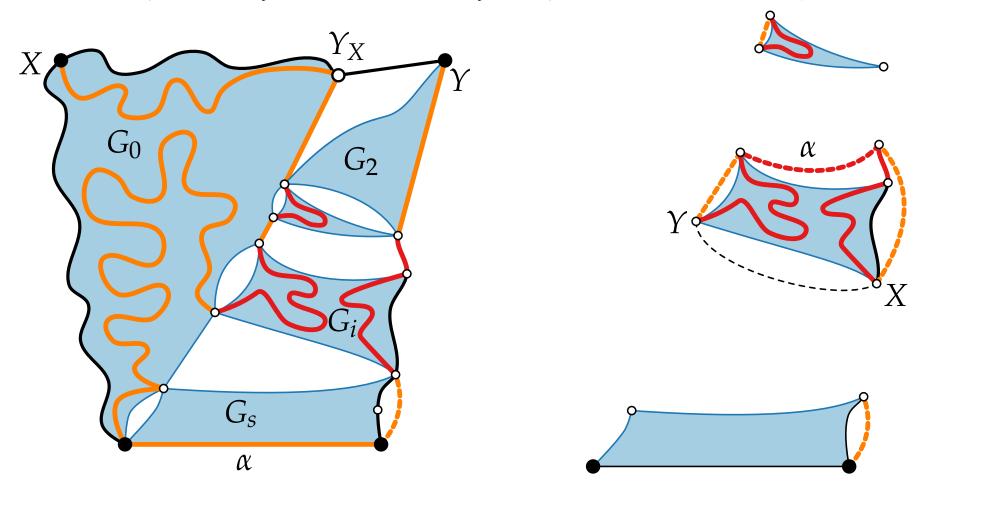


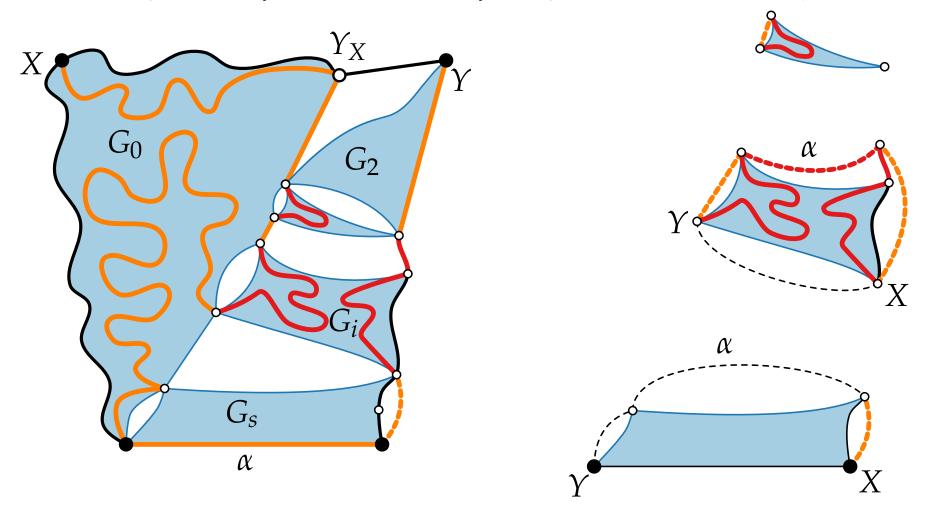


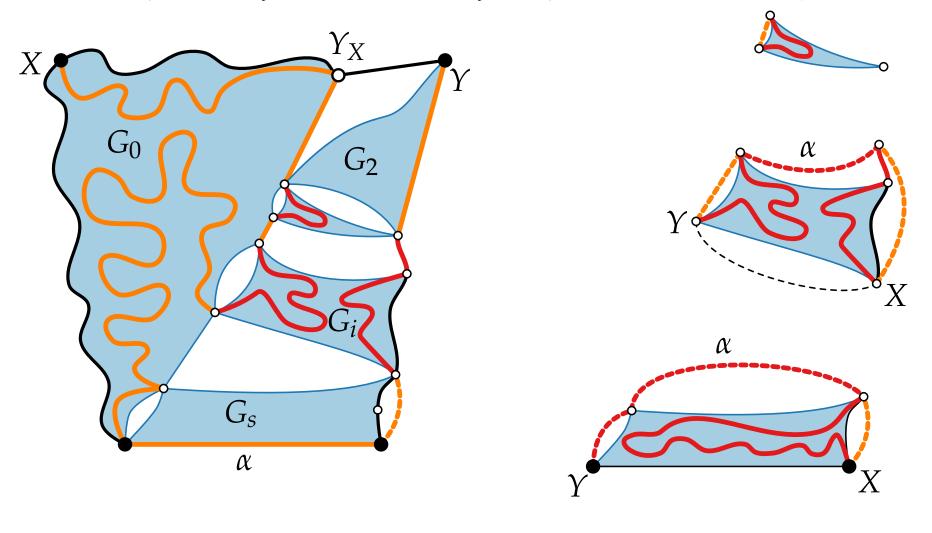


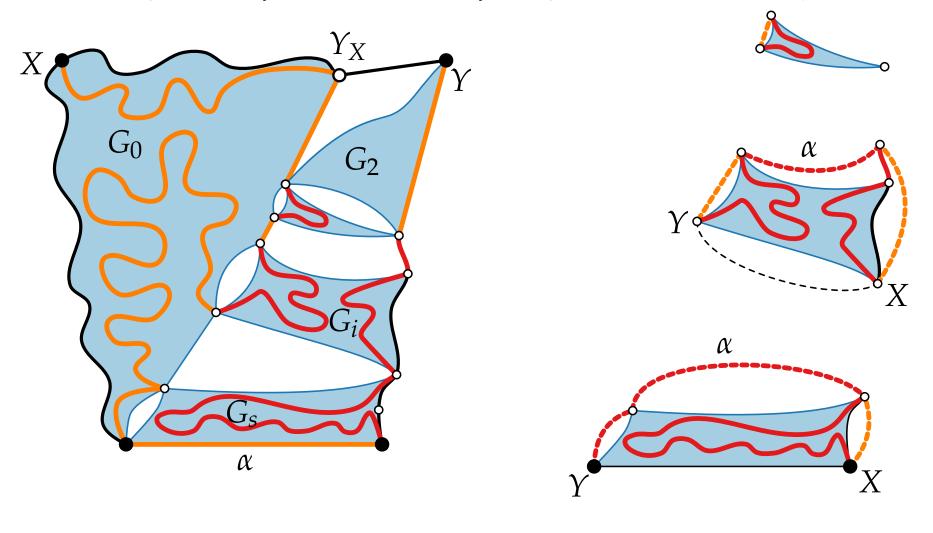


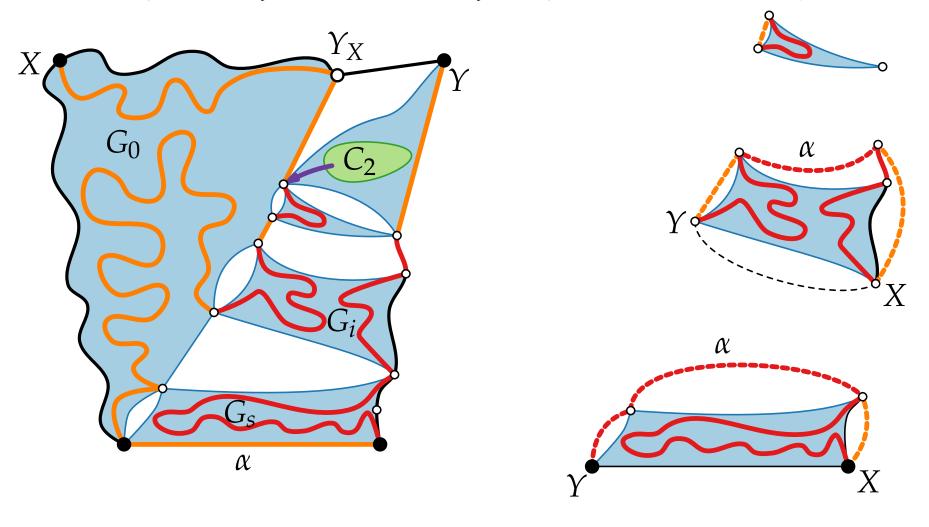












5 Linear-time complexity for 3-connected graphs

We now explain how to find the $T_{\rm inc}$ -path in linear time for 3-connected graphs, i.e., prove Theoremi2. We first argue how to bound the time spend on recursions, once we know which case applies and have found the subgraphs. Clinic is the easier part. Descript work or store cutting parts (for determination of cases) and how to store adjacencies to sides (for determination of the necklese); these data structures are not complicated, but arguing that their updates take overall inner time is lengthy.

As we notations will be useful. First, set $D_T := \sum_{(I \in P)/P} \deg(I)$; we aim to show that the running time is $O(P_T)$. Next, set $D_T := \sum_{(I \in P)/P} \deg(I)$, where V(P) are the vertices of P. Since every vertex has an incident interior face, we have $P_T = D_T = I$ thence suffices to step as running time of $O(P_T + P_T)$. Next, set G to be the set of all subgraphs that we used in some recursion. Let G^* be the graph that we would get if we nestered into D is some recursion. One that G^* is still a red got ball wave used as virtual eight some recursion. One that G^* is still a red got ball wave used as virtual eight is some recursion.

Baham graph. Let V_X be the set of vertices of G that were exterior in some subgraph $G' \in G$. Note that, when recursing into G', we obtained a Tutte path of G' that visits all exterior vertices in G'. When combining Tutte paths of subgraphs, the resulting path always visits the same set of vertices. So $V_X \subseteq V(P)$, and in particular

 $|E_n| \le 3|N_n^2| - 6 \cdot G(D_r)$. For any subgraph G of G is $E_n(G)$ be the edges of G' that are exterior in G', but were not exterior (or dist sot exist) in the power graph that we recovered from North test $|E_n(G')| \ge 1$ in all cases. These where G is not between the case or is find the archives) is $O(|E_n(G')|) \le 1$. In other interest one or to in first the archives) is $O(|E_n(G')|) \le 1$. Of the order order of the order order of the order order of the order order of the order or

resume that with the planar embedding we obtain a standard data structure with the following: Every vertex x knows whether it is exterior, and has a list L(x) of its incident faces and edges in order. This list is circular if x is interior, and begins and eads with the exterior edges at x other (There are exactly two such edges since all graphs in G are 2-connected.)







Now, consider the one where we can an ordin side S_c , except that g' and (m, g'), and already have been discincent to the side and then do not get exament (H_{c_1}) , was not by ve 0. So, then all ove extrins due to it should be appended to $P(I_c)$. If I(f) was not yet on S_c , then all new extrins due to it should be appended to $P(I_c)$. If I(f) was not yet on S_c , then all new extrins due to it should be appropriated to S_c . If S_c only so that the new case of S_c is the new tools be a simple size of S_c . If S_c only S_c is a size of S_c is the new case in the new critics. If B_c $P(I_c)$ precisionly was empty, then we can simply scan. So by using the correct of the above approaches, we can create the coverex does $P(I_c)$ as long as we seem the following.

Property 4. Assume that, when recursing from G to a subgraph G', we scan an entire side S'_i . Then either

5.4 Details for the individual cases

We now finally so through all cases and fill in some case-dependent details. In particular, we must explain We now finally go through all cases and fill in some case-dependent details. In particular, we may consider the trace applies, and of the property of the pro

- In all cases, we continue with the notations and assumptions that were used in Section Σ .

 (A) (ii) is straightforward to text whether Case [] applies, since we know the conter face.

 (b) if it is straightforward to text whether Case [] applies, since we know the content face.

 (c) (iii) is straightforward to text whether Case [] applies, since we have the content face.

 (c) $G = C_0 \cap (\Omega(Y))$, i.e., we remove the edge that was the right of C or C

- (D) There are no divided faces.
 [E] Graph C had no cutting pairs, so all P-lists as empty. Hence Property ⊕bods.
 (F) The only vertices that could be new to a side are the ones that shared a face with (W,Y) in G. We should therefore exam along the entire side S_{right} of G ∩ All vertices in S_{WY} \ \(\begin{align*}{c} W, Y, \end{align*} \) are different from W,Y, hence new to S_{right}.
- † "right" is a shortcut for "the index i such that $W = c_i$, i.e., side S_i' is the right side". Since this can be looked up in O(1) time from the location of X and Y in C, we use this convenient shortland for this and the other sides.



- its other end. \bullet Vertices, edges and faces are cross-linked, i.e., each vertex/edge/face knows all its occurrences in the lists $L(\psi)$ and/or L(F). It imports that the first $L(\psi)$ and/or L(F). It imports that the first and will not be explained here; see, for example, $\Xi \Pi \Xi$, find cutting pairs and leftmost necklaces efficiently, we need to store more information as follows, see

In this cutting pass and lettinos incidence efficiently, we need to store more internation as follows are C. We store a circular like of for internations $\alpha_{11}, \alpha_{12}, \alpha_{13}$ to these orders in K. We then the contrast of C. We store a circular like of α_{11} to the contrast of α_{12} to the contrast of α_{12} to the contrast of α_{13} to the contrast

Case gift (Fig. 2).

Case gift

Since we can along units engineetic sides. Properties \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} (a.g. \overline{g}_{ij}) to the visible. Case \overline{g}_{ij} engineetic sides in the first g_{ij} becomes \overline{g}_{ij} and g_{ij} becomes \overline{g}_{ij} becomes \overline{g}_{ij} and g_{ij} becomes \overline{g}_{ij} and g_{ij} becomes \overline{g}_{ij} and g_{ij} becomes \overline{g}_{ij} and g_{ij} becomes \overline{g}_{ij} becomes \overline{g}_{ij} and g_{ij} becomes \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline{g}_{ij} becomes \overline{g}_{ij} and \overline{g}_{ij} becomes \overline

initialize the necklace N with \langle outer face, $Y_X \rangle$ $\langle f_{peer}, x \rangle \leftarrow$ first two entries in $L(Y_X)$ append f_{prev} , x to N while $x \neq U$ do

hillo $x \neq U$ do

If x is not face-adjacent to U then

foreach face f_{perc} indicident to x, starting after f_{perc} in cew

foreach vertex x' on f_{perc} , starting after x in ecw order

If x' is face-adjacent to S_{nijh} then

append f_{perc} x' or f_{perc} f_{perc}

 $f_{next} \leftarrow \text{first face in } V(x, \text{bottom})$ if f_{next} does not contain vertex of S_{right} then $append f_{next}, X \text{ to } N$ else // to be leftmost, include all vertices along $f_{\rm sext}$

(b) The path P_i Figure 3 (a) Case 2 (b)-(d) proof of Lemma 1 for Case 2 (repeated from page 5)

(c) The path P_b

Algorithm 1: scan_side(i, c, d)input: c and d are vertices on the outer face, the ccw path from c to d is on side S_i put: c and d are vertices on the outer face, the cew path from e to d is on side, reconcivertex v on side, S_i from v to d is d. If v was not on side S_i before then foreach interior face f in L(v) in clockwise order after the outer face) do add v to L(v) in clockwise coder after the outer face) do L(v) if f was not incident to side S_i before then L(v) and L(v) is L(v) in L(v) i Algorithm 2: scan_face(f, v, i)input: f is an interior face and incident to exterior vertex v on side S_i put: f is an interior use and incodes to exterior vertex vreach vertex $w \in L(F) \setminus \{v\}$ in clockwise order after v do if w is interior then $\lfloor u \rfloor$ append f to V(w, i)

P. For any two distinct isides S_i, S_j we store a list P(i,j) of interior faces that contain a cutting pair $\{v,w\}$ of G_i with v on side S_i and v on side S_j . This list is sorted by the order in which there faces are incident to side S_i ; are $P[S_i]$ S_i . Note that P[i,j] is the reverse of P(i,j), but it will be convenient to store both of them. We assume that any face f faces and all the P[i,j], the size only a constant overhead questions.

Initialization. We agree how to initialize the data structure in O(n) time; this then also shows that it was given interest to the correct, and breas initialize U washing the contract of the structure of the contract of the structure of th

case some case-dependent details. Assume that we recurse into some subgraph G' of G. V will use "primed" versions (such as V' and S') for the data structures and properties of G'.

 $\text{Figure $\frac{1}{N}$ Case $\frac{1}{N}$ (a) G_b^+ satisfies $c3c(X,U,W,y)$, (b) Case $\frac{1}{N}$ (c) Case $\frac{1}{N}$ (d) Case $\frac{1}{N}$ (repeated from $\frac{1}{N}$) $\frac{1}{N}$ (a) $\frac{1}{N}$ (b) $\frac{1}{N}$ (b) $\frac{1}{N}$ (c) $\frac{1}{N}$ (c) $\frac{1}{N}$ (d) $\frac{1}{N}$ (d)$

was adjacent to the bottom side, then this must hence have happened at W_v which makes $\{y,W\}$ a cutting pair. By our sorting of P(tight, top) hence $w=W_v$ which masses that f_v^a then is also adjacent to the bottom side at w. So f_v^a retains all side incidences that f^a bad. As for f_v^a , this is involved in a recursion only if Case [Eapplies (due the becomes part of the outer face). Here, the corners of G_v^a become V_v^a by which means that face f_v^a touches all four sides since it is

Each case will state which corner Z' of G' corresponds to which corner Z of G, i.e., takes the entry of Zin G; this defines corresponding sides. We will do this so that the following holds.

Property 1. Assume that a vertex v is incident to a side S_i in G and belongs to some subgraph G' that we recurse in. Then v is incident to the corresponding side S' of G'.

recave in. Then v is incident to the corresponding side $S_v \in G$. We cannot staffer to find $v_v \in S_v \in$

overall.) Subgraph G' inherits the lists V_i , i.e., V(w, i) = V(w, i) for all vertices w that are in G', and no time is spent on creating these since w simply keep the same lists. We need to argue that this does not create fade positives for an interior vertice w of G. Recall that $V(w_i)$ shows interior fixes G is the contain w as well as a vertex on S_i . Charge, f all contains w in G', and it does contain a vertex on the corresponding side S_i' of G because of the bilimiting property of our cases:

Property 2. Assume that an interior face f is incident to side S, in G, and f (or some part of f obtained by dividing f along a wirtual edge) is an interior face of some subgraph G that we recurse in. Then f is incident to the corresponding subs S_i of G.

one contains to the corresponding size S_i of C. That was not divided, because then all vertices of f must also belong to G_i and by Property [] so that even remains an all sides that it was on in G_i . This property is not Subgraph G_i and such insteads the sides of G_i and G_i

y constant space overhead.) The initialization of the P-lists for G' will depend on each case, but as will be seen, they are either empty inherited from G (with minor modifications). Furthermore, all cutting pairs of G that also exist in G' get

Secondary (parts of) piles. We have initialized the data structure of G' to that it has no faite positive centric that should be the best, but it may be mining some critic since some vertice super bows to a side. To add these mining centric, we scan (parts of) each side S_i in such a way that all vertices that are now of S_i see guaranteed to be assumed, i.e. we discuss a discuss i of the some i (G_i). All some one vertices G_i of S_i find cone listed below will explain exactly what needs to be examined. We should insurial here G_i of S_i (S_i find cone listed below will explain exactly what needs to be examined.) We should insurial here G_i of S_i (S_i find cone listed below will explain exactly what needs to be examined.) We should insurial that the lates. To make that feasible, we now that G_i one and attriction, and that the following shole in our

Property 3. Assume that, when recursing from G to a subgraph G', we scan along a part S'_{cd} of some side S'_c . Then, one of the following holds:

(i) We scan the eather side $(\ell, \ell, c) = \epsilon'_1$ and $d = \epsilon'_{\ell+1}$), and the vertices between c and d (if any) were not

Figure Case 4 (a) A simple interior U-necklace that is not leftmost due to face f (which yields a cuttin pair $\{x_j, x_i\}$), and since it could include vertex z. (b) The graphs G_1, \dots, G_{ν} . (c-d) Case [a] The path Pafter using the substitution trick and (d) assignment of the representatives. (repeated from page 1)

points, so we find the behavior II tracelators. We come that the time spent for updating the data structures for graph G_2 later. First, we spend O(1) time pre-vertex that we found, hence O(1) time receil. Since G_1 has we eved spen of the order face, this is accounted for the $O(1/k_1)G_2$ bright was well-varied vortex of the size of the $O(1/k_1)G_2$ bright was described by consider for G_2 . Second, we spend $O(\log H_2)$ lines on seaming an interior face f includes the solution of the size of

angent to the edge one can hence we age is caused curring the spane. For Case \mathbb{R}_{+} indical, the off-case is a finite of the case is the off-case indical, except that we search from the top downward, ruther then from the bottom quench, which requires exchanging 'clockwise' by 'ove' in the order of the case of t



As let f_i , this is involved an a continuous η_i if Cose Lappajos (see it becomes part of the order benchmark) is included to g and g becomes (P_i, q) , which means that fair f_i from the all form sides more it is included to g and g. The cose is g is considerable and g is a region of g is the state of the P-blats in G. Pileth, right) was empty, else C me. We would have applied, when G is the GCase 3 If Case does not apply, then we test for Case by taking the last element f^* of $\mathcal{P}(\text{left}, \text{top})$ and checking whether it contains X. The treatment of this case is symmetric to Case \square

incidence. Assume most that f has parts in graph G_i for some $1 \le i \le s$, and $f_{i-1} = t$. We recurse in G_i only if we apply the substitution trick, in case of which edge (f_{i-1}, x_i) is removed. In particular, the face f_i that we say this description, f_i is an interpolar of G_i of G_i (G_i , G_i , G_i) and G_i , G_i in the graph G_i = G_i (G_i , G_i , G_i), G_i , G_i ,

of the left corners s_{i-1} and s_i) is incident to all flour sides of G_i^0 . In particular, now side incidence leaves leave in an H_i . Here, has given in graph G_i be some G_i in G_i and G_i in the particular implies that we are in Case[H]. We sagne that the side incidences are preserved for the resulting graph G_i^0 (i.e. they deep visits have like G_i^0 of G_i^0 in G_i^0 in the G_i^0 of G_i^0 in G_i^0 in the particular in G_i^0 in G_i^0 in G_i^0 in the G_i^0 of G_i^0 in G_i^0 in the G_i^0 in $G_i^$

the edge, a contribution, by cannot be edge (t_{i-1},t_{i-1}) and not reconstruct that f was directly as (t_{i-1},t_{i-1}) , in case of which we only need to worry if f had a vertex z on the bottom side of G. If z = U, then (z_i,t_{i-1}) would be a cutting pair by $t_{i-1} \neq t_i$, and we would have applied $\text{Cose} \supseteq S_i = z = W$, which implies $t_i = W$ (else we would have a cutting pair within the right side of G). So f contains t_i , and there is incident to all sides of G^* .

Assume that f was divided by (x_i, t_i) , in case of which we only need to worry if f had a vertex z on the top side of G. We claim that f contains Y. Namely, if $z \not\in Y$, then $\{z, t_i\}$ would be a cutting pair by $t_{i-1} \not\equiv t$. Since we did not apply case U herefore f contains Y. This implies $t_{i-1} \not= Y$ (see we would have a cutting pair within the right side of G). So f contains t_{i-1} , and hence is incident to all

With this, we have explained how to do all steps in all cases. Note in particular that necklace takes no more time than what was spent on scanning vertices and faces. Note further that Property holds, hence the running time is linear (in the degrees of vertices and faces touched by the output path P (ii) $d = c'_{i+1}$, (c, d) is an edge, c was on S_i in G, and d was not on S_i in G.

The worth pointing for that the second of the state of the same of the state of th

we will scan at w from sore $\mathcal{F}_{\mathcal{F}_{\mathcal{F}_{\mathcal{F}}}}$ into use comma, some time is no need to some at v.

We likewise exempt in line $|\psi_{\mathcal{F}_{\mathcal{F}_{\mathcal{F}}}}|_{\mathcal{F}_{\mathcal{F}_{\mathcal{F}}}}$ for g from scanning if it was already on the side, i.e., if the list $\mathcal{F}_{\mathcal{F}_{\mathcal{F}_{\mathcal{F}}}}(i)$ in G was non-empty. As above, one argues that this will mean no missed entries in the data structure. Avoiding these face scans is the crucial insight to being the running time down to linear.

Lemma 7. The total time to scan sides of subgraphs of G is $O(D_V + D_F)$.

Proof. When scanning sides (for one subgraph G'), we may spend time on vertices that end up not bein scanned, due to $\lim[\overline{G'}]$ By Property \overline{B} , there are at most two vertices on each side, hence O(1) in total We count this as overhead to the $\lim \overline{O(B_c/G')}$ that we budgeted for G' earlier, and do not consider

We count this as overhead to the time $\widetilde{O}(E_{s}/G^{\prime\prime})$ that we badgeted for $G^{\prime\prime}$ cultile, and do not consider it for the time in the state here. So that the state of the state is a same of the state in the state of th

that this age spin errors, and more task $\geq \log(f)$)—since it is even by good part in $\log(f)$. Consider some part f of [f] cooling f = f) that give its cannel, which takes $f(\log(f))$ time. Let $f_{1,\dots,f}$, (with $g \in \{1,\dots,d\}$) be the pieces of f that belong to f. Therefore $\sum_{i=1}^{n} \deg(f_{i},f) \geq \deg(f)$. So it affices to according to Glove f_{i} , f) where f is considered as f is a finite part of f in the part of f is a finite part of f in f in f is a finite part of f in f

Maintaining the correct order, During side, seat (i, c, d), we possibly side arriers to $F(I_I)$, $V(w_i)$, and $F(I_I)$, $F(w_i)$ and side S_i . The Telline are containt the not particular and $F(I_I)$ for some side of S_i and S_i . The Telline are containt to an in particular is created for shoring this: we usually some nearly the whole side.

Assume first that we can an edge (c, c_i) , where C_{i+1} is never to take S_i . Therefore, the existing entries in $V(w_i)$ $\alpha \in F(i_i)$ we flow so we can a very low of S_i , where C_{i+1} is never to take S_i . Therefore, the existing entries in $V(w_i)$ $\alpha \in F(i_i)$ we flow where C_i is not part to take S_i . Therefore, the uniform S_i is the containing of S_i and S_i is the simple approach and we maintain the distribution of S_i and S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and we maintain the distribution S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and we maintain the distribution S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and we maintain the distribution S_i is the simple approach and we maintain the distribution S_i is the simple approach and we maintain the distribution S_i is the simple approach and we maintain the distribution S_i is the simple approach and we maintain the distribution S_i is the simple approach and we maintain the distribution S_i is the simple approach and we maintain the distribution S_i and S_i is the simple approach and S_i is the simple approach and S_i is the simple a

Algorithm 3: necklace_scan_4a in the increase r_i was r_i was r_i was r_i and r_i with the formula for f_{im} in clockwise order do formula for f_{im} in clockwise order do formula for r_i or f_{im} , in clockwise order do if r_i is face adjusted to r_i then r_i in clockwise order do if r_i is face adjusted to r_i then r_i in clockwise order do if r_i is face adjusted to r_i then r_i in $r_$ else // Have reached the face of N incident to X. $f_{lext} \leftarrow \text{last face in } V(x, \text{top})$ if f_{lext} does not contain Y then $\text{append } f_{lext}, X \text{ to } N$ else // to be leftmost, include all vertices along from se // to be lettmost, include all vertices all repeat $x' \leftarrow \text{vertex after } x \text{ in clockwise order on } f_{\text{next}}$ append $f_{\text{next}}, x' \text{ to } N'$ until x = X

Note that, in both cases, once we found the neithice $(x_0, f_1, x_1, f_2, \dots, f_n, x_n)$, we can also easily find the vertices f_1 for $i = 1, \dots, x_n = 1$ for and f_n are determined from the case). Namely, to find f_n by f' using the one that coorse can be considered from the case). Namely, to find f_n by f' using the one that coorse can later on the outer four of three zero. This gives all the required information to determine the subgraphs to recurse in f_n for the subgraph of f_n of the constant f_n for f_n f_n fo

ield L_{ii} take the place of V in C (with this, side N_{iij} , of G, corresponds to side N_{iij} of G are required.

As always, corresponds to the side where textical our expelled regarded a place in C are still on as to ministation the ever order in C.

We word to agree that any face f that was divided in Case [alone, pa hose my side intriduces, the contraction of C is contracted to C in the contraction of C in the contraction C i

Store:

Store:

• corners

Store:

• corners

• faces: all vtcs on each side

Running Time

Store:

corners

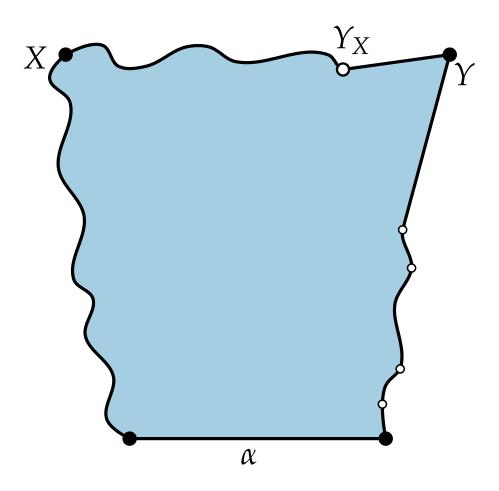
• faces: all vtcs on each side

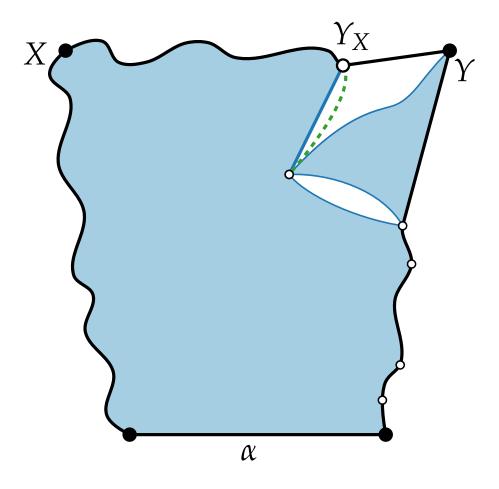
• vtcs: all face-incidences to each side

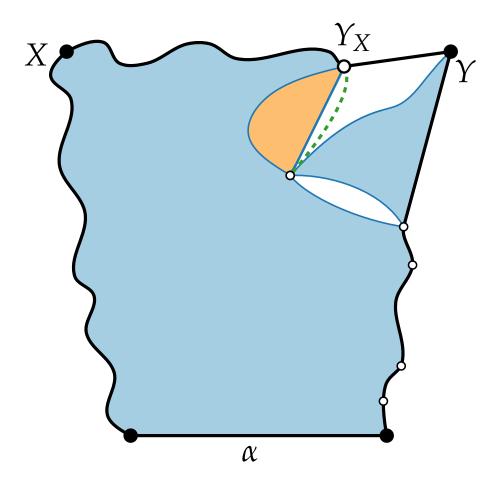
Running Time

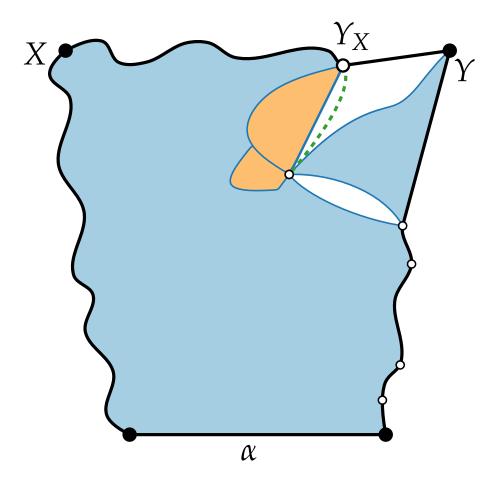
Store:

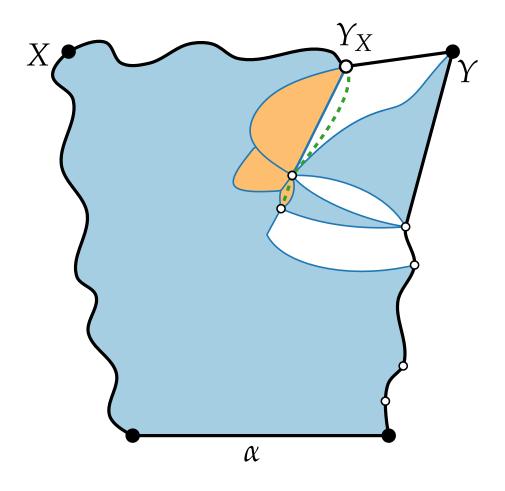
- corners
- faces: all vtcs on each side
- vtcs: all face-incidences to each side
- sides: all cutting pairs

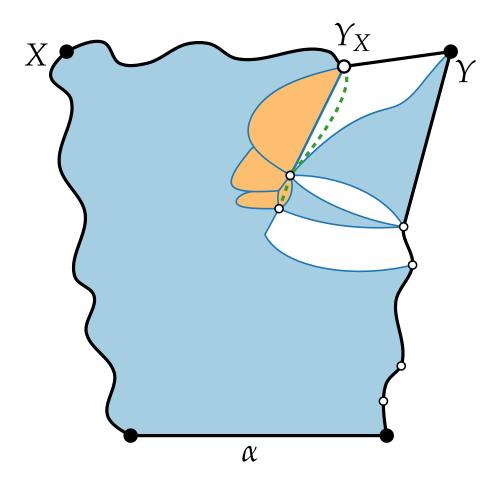


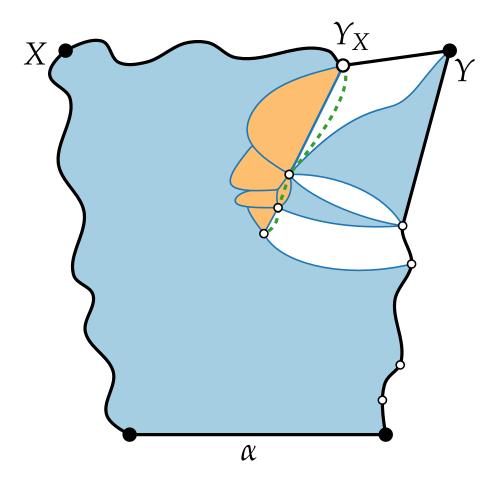


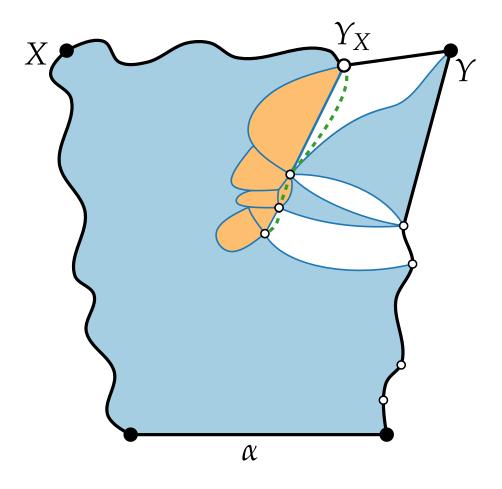


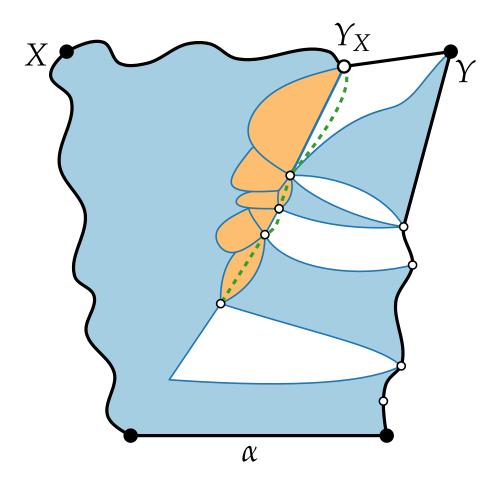


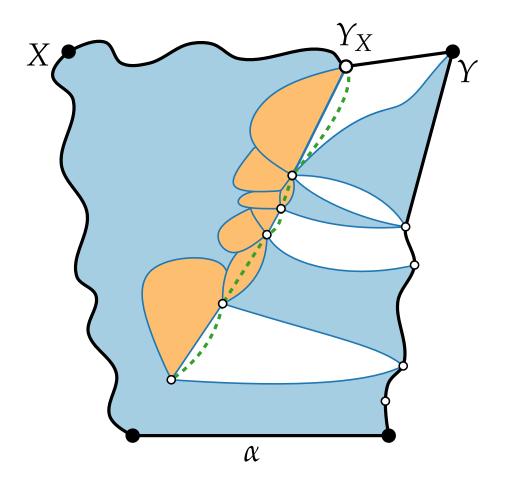


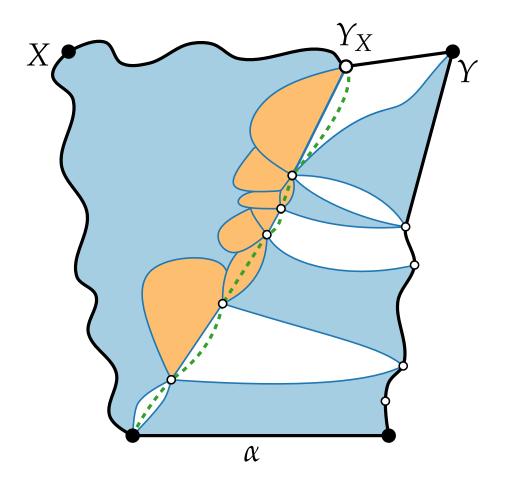


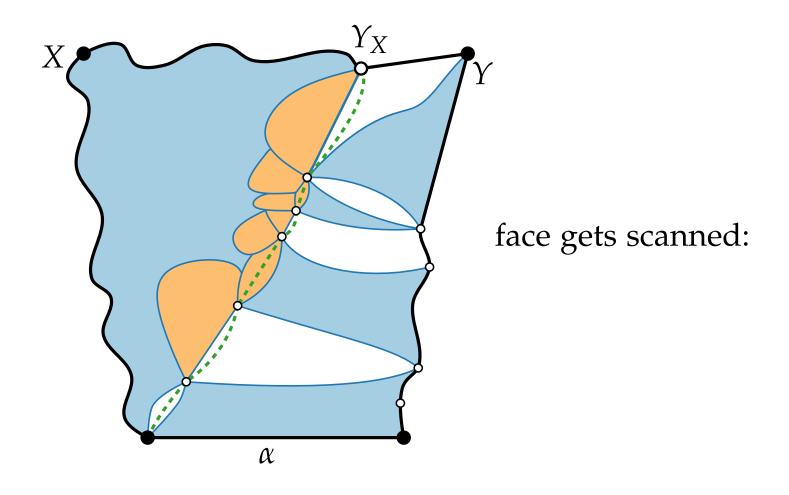


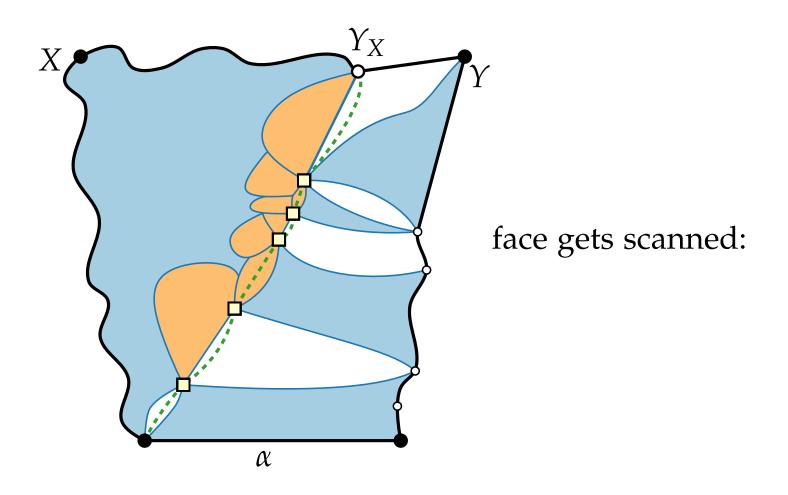


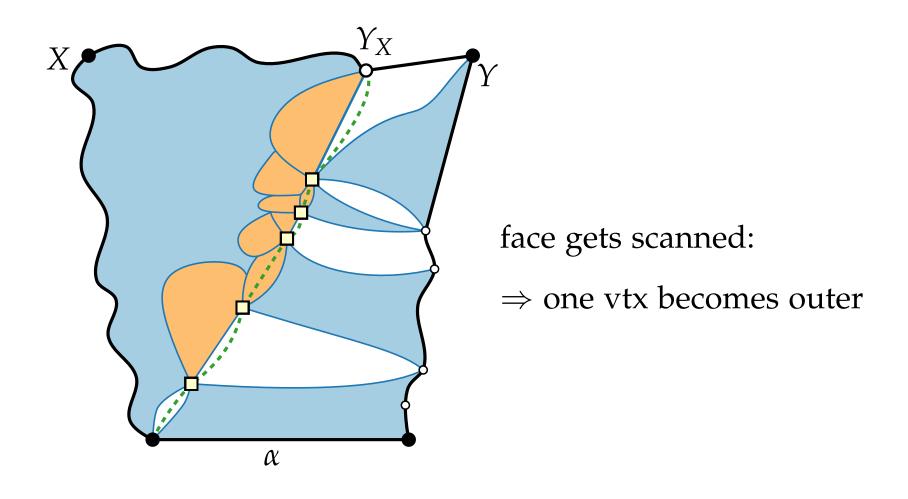


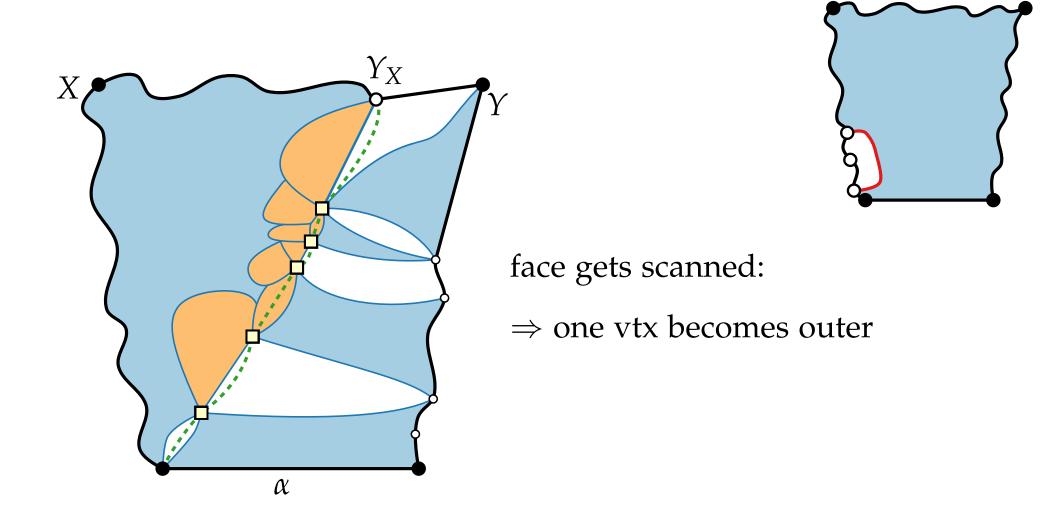


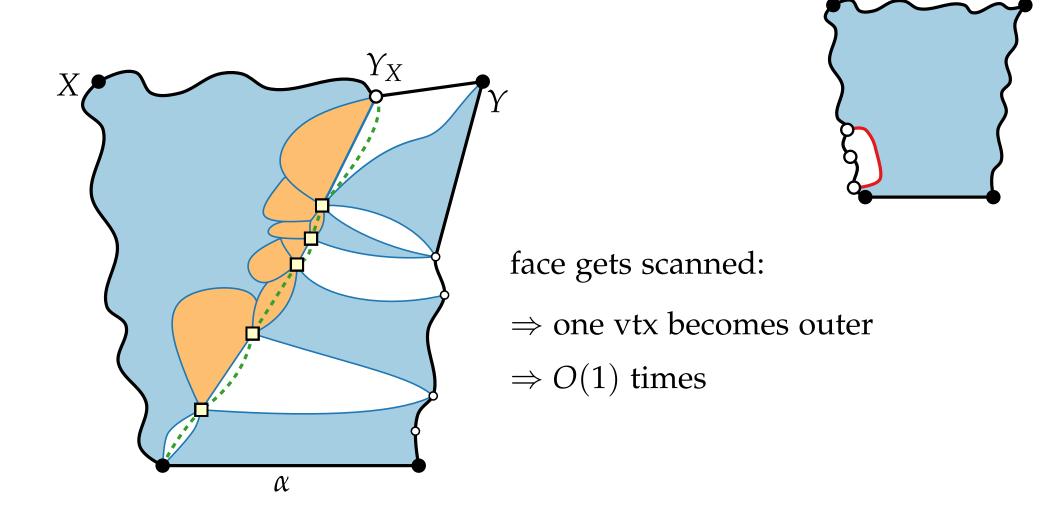


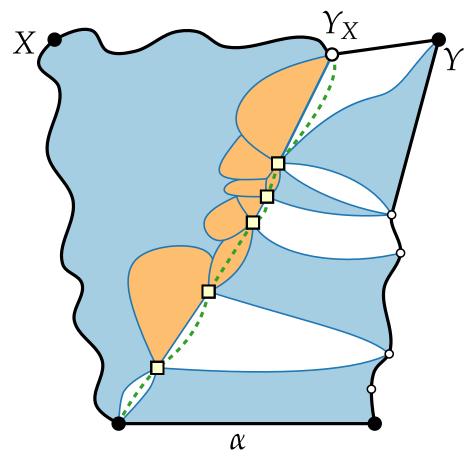


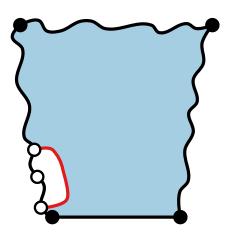






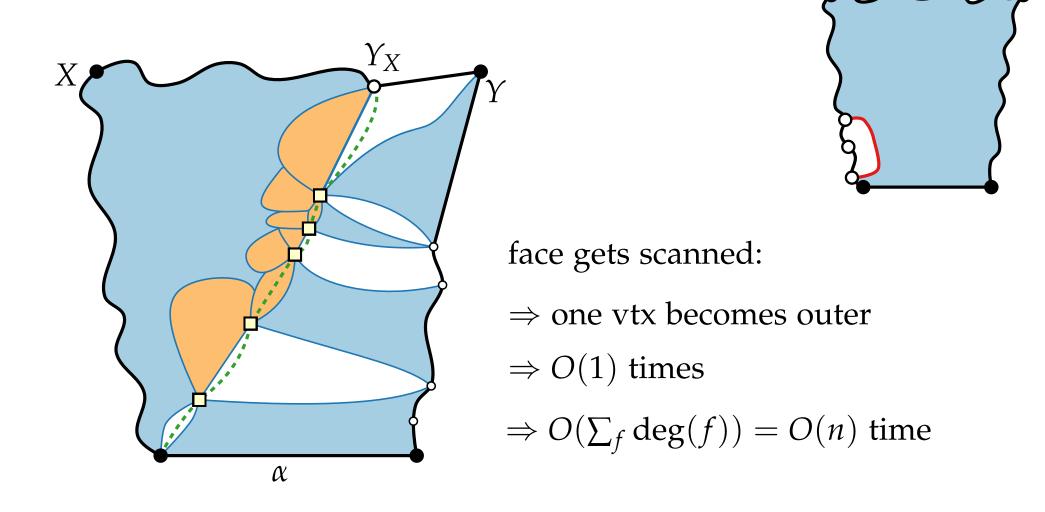






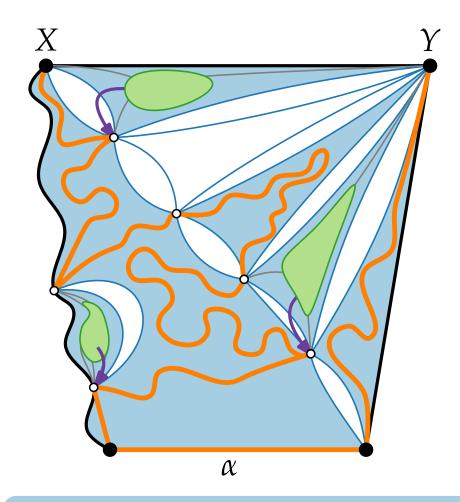
face gets scanned:

- \Rightarrow one vtx becomes outer
- \Rightarrow O(1) times
- $\Rightarrow O(\sum_f \deg(f)) = O(n)$ time

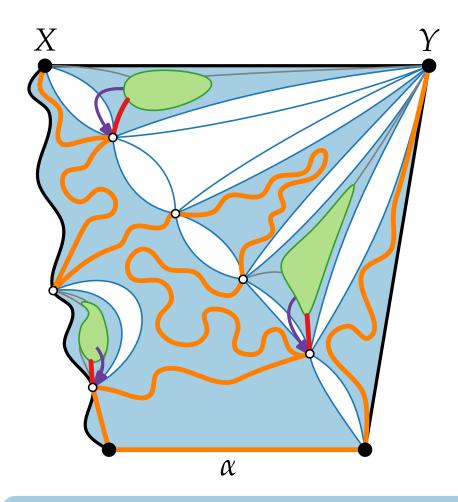


Theorem.

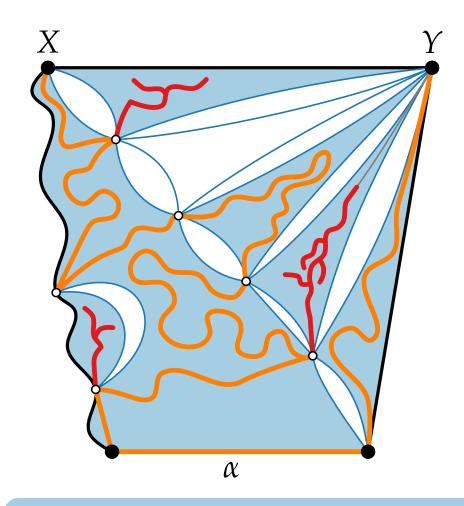
G int. 3-conn., X, Y, α on outer face \Rightarrow T_{int} -path in O(n) time.



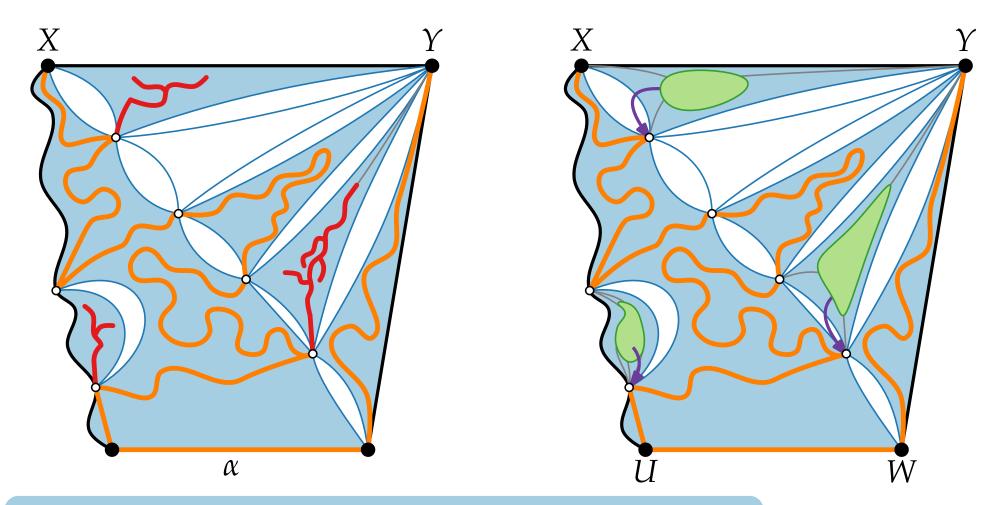
Theorem.



Theorem.



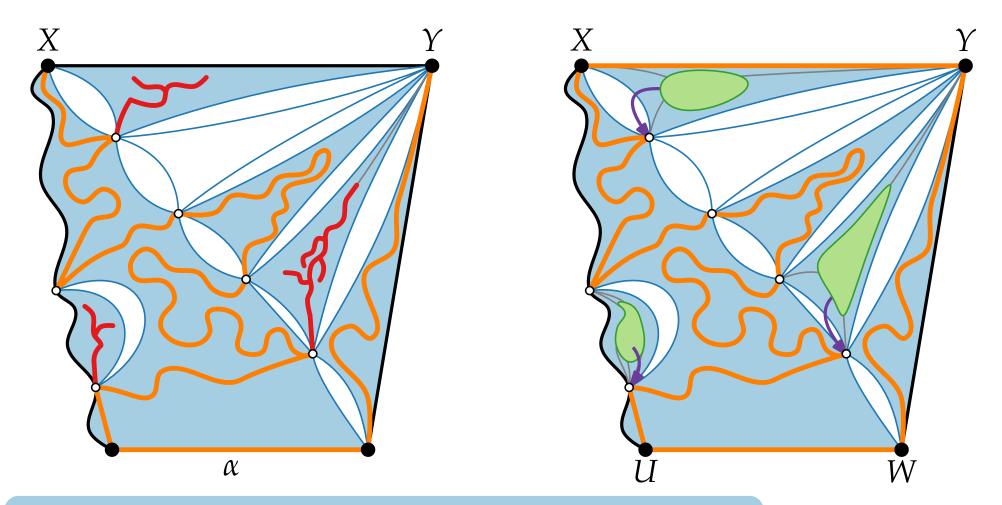
Theorem.



Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.

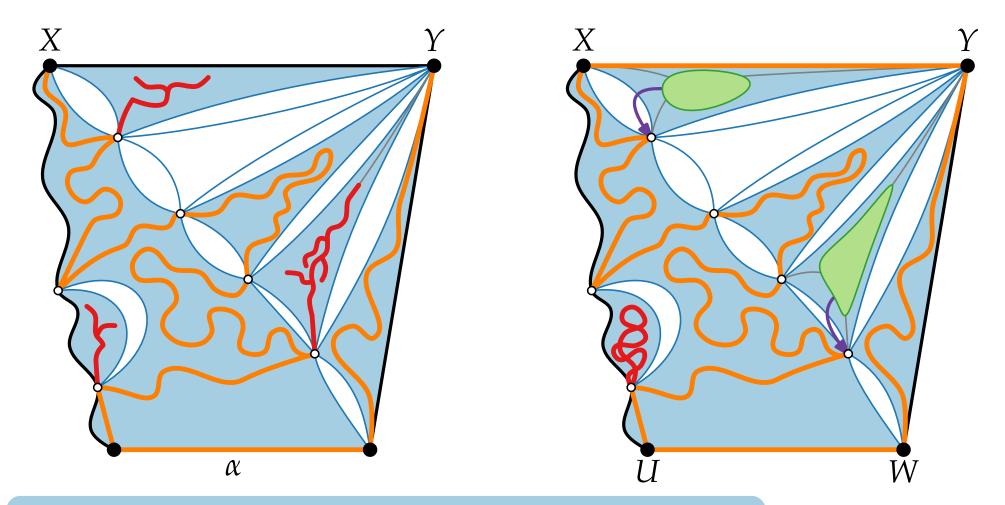
Theorem.



Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.

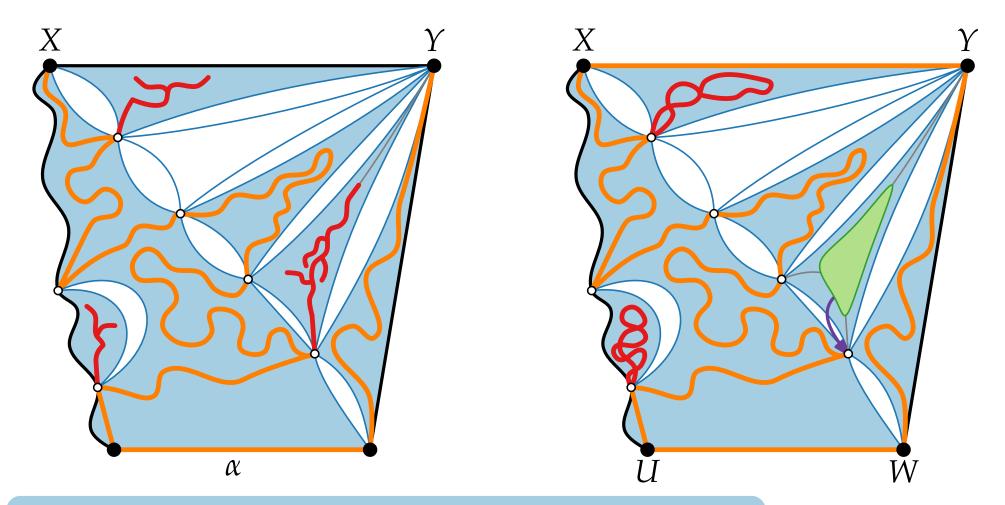
Theorem.



Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.

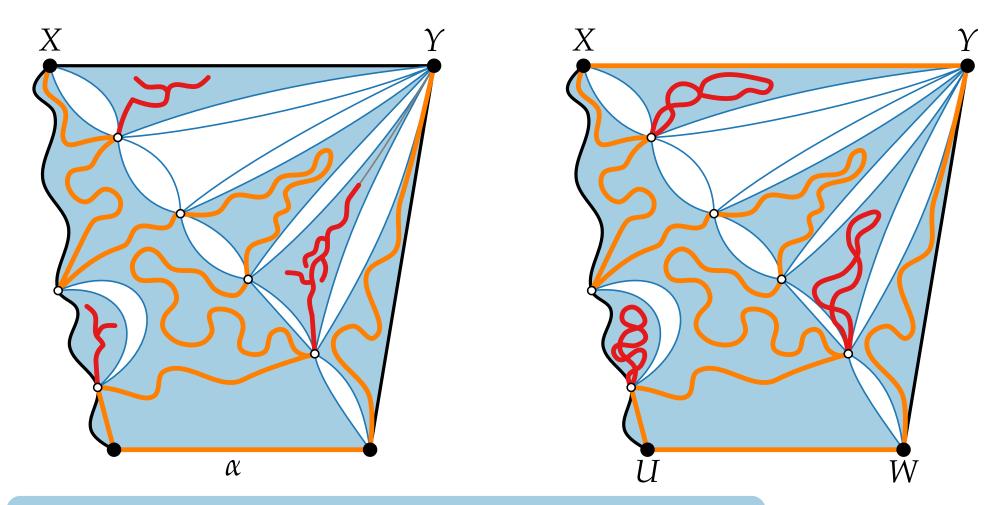
Theorem.



Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.

Theorem.



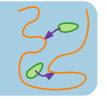
Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.

Theorem.

Theorem.

G int. 3-conn., X, Y, α on outer face \Rightarrow T_{int} -path in O(n) time.



Theorem.

G int. 3-conn., X, Y, α on outer face \Rightarrow T_{int} -path in O(n) time.



Theorem.

G 2-conn., X, Y, α on outer face \Rightarrow Tutte path in O(n) time.



Theorem.

G int. 3-conn., X, Y, α on outer face \Rightarrow T_{int} -path in O(n) time.



Theorem.

G 2-conn., X, Y, α on outer face \Rightarrow Tutte path in O(n) time.



Theorem.



Theorem.

G int. 3-conn., X, Y, α on outer face \Rightarrow T_{int} -path in O(n) time.



Theorem.

G 2-conn., X, Y, α on outer face \Rightarrow Tutte path in O(n) time.



Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.

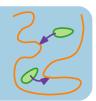


Theorem.



Theorem.

G int. 3-conn., X, Y, α on outer face \Rightarrow T_{int} -path in O(n) time.



Theorem.

G 2-conn., X, Y, α on outer face \Rightarrow Tutte path in O(n) time.



Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.



Theorem.

G int. 3-conn. \Rightarrow 2-circuit in O(n) time.



X, Y, α on different faces?

Theorem.

G int. 3-conn., X, Y, α on outer face \Rightarrow T_{int} -path in O(n) time.



Theorem.

G 2-conn., X, Y, α on outer face \Rightarrow Tutte path in O(n) time.



Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.



Theorem.

G int. 3-conn. \Rightarrow 2-circuit in O(n) time.



X, Y, α on different faces?

Non-planar graphs?

Theorem.

G int. 3-conn., X, Y, α on outer face \Rightarrow T_{int} -path in O(n) time.



Theorem.

G 2-conn., X, Y, α on outer face \Rightarrow Tutte path in O(n) time.



Theorem.

G int. 3-conn. \Rightarrow binary spanning tree in O(n) time.



Theorem.

G int. 3-conn. \Rightarrow 2-circuit in O(n) time.



X, Y, α on different faces?

Non-planar graphs?