Motion Planning

O'Rourke, Chapter 8

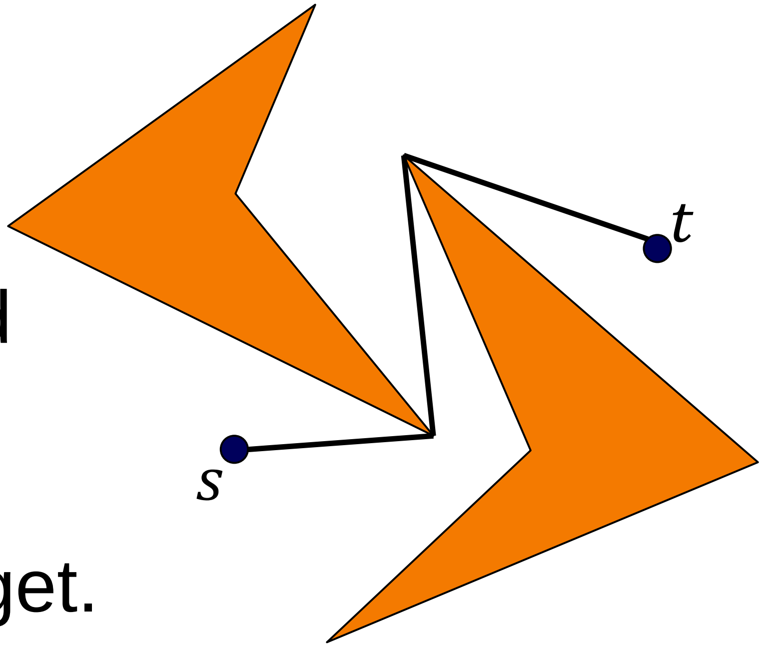
Shortest Path

Goal:

Given a collection of disjoint polygons in the plane, and give a source s and target t, find the **shortest** path from s to t that avoids the polygons.

Key Idea:

Since the path is shortest, it only bends at corners and will otherwise consist of straight segments between vertices and the source/target.



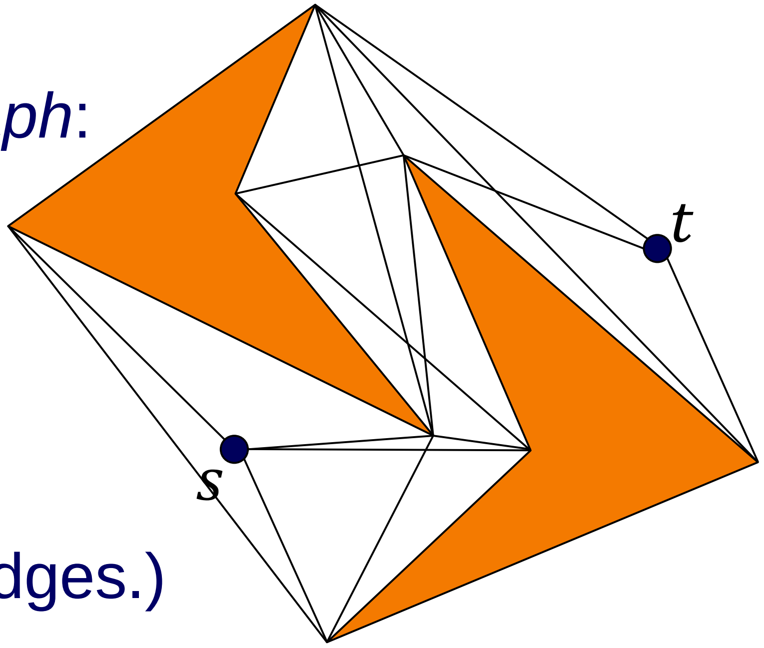
Shortest Path

Goal:

Given a collection of disjoint polygons in the plane, and give a source s and target t, find the **shortest** path from s to t that avoids the polygons.

Approach:

- Construct the *visibility graph*: A graph whose nodes are vertices of the polygons plus s and t, with edges between nodes that "see" each other. (These include polygon edges.)



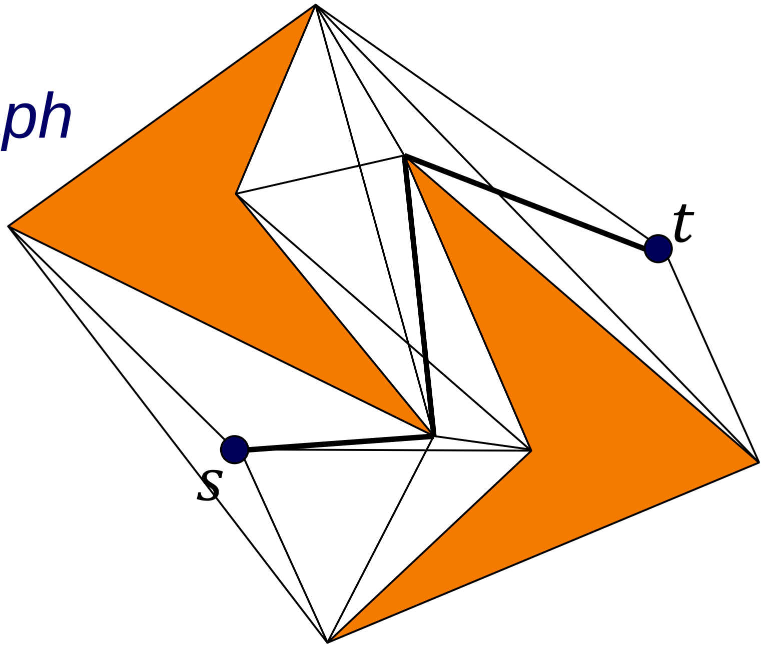
Shortest Path

Goal:

Given a collection of disjoint polygons in the plane, and give a source s and target t, find the **shortest** path from s to t that avoids the polygons.

Approach:

- Construct the *visibility graph*
- Find the shortest path from the source to the target in this graph.

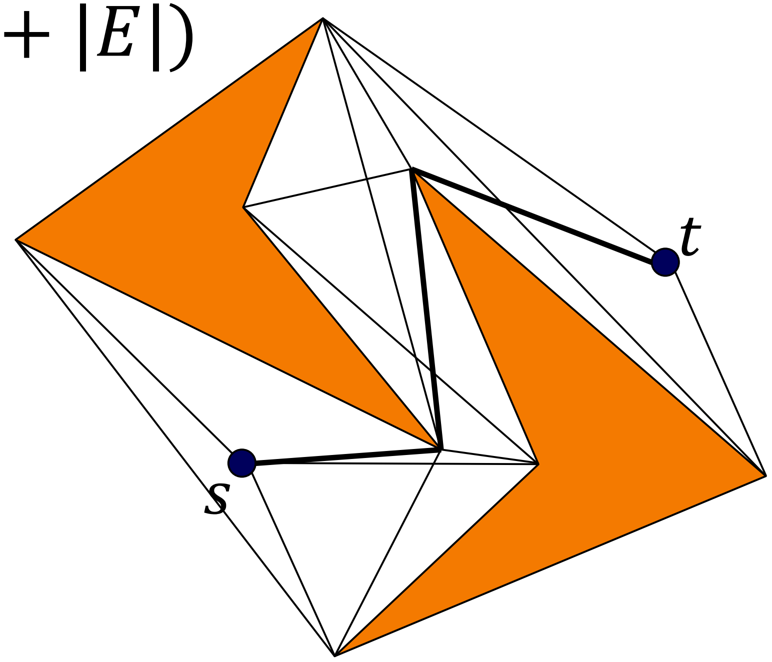


Shortest Path

<u>Find the Shortest Path</u>:

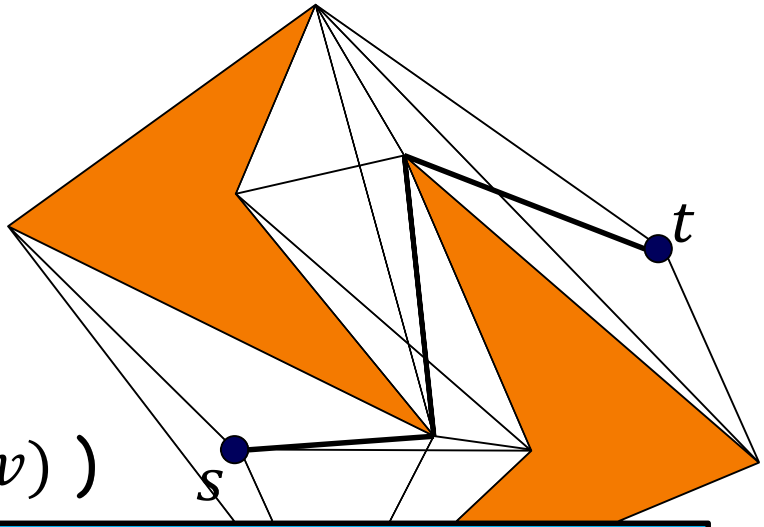
Setting the weight of each edge to the Euclidean distance between its endpoints, this can be solved using Dijkstra's Algorithm.

⇒ Complexity: $O(|V|\log|V| + |E|)$



Dijkstra's Shortest Path Algorithm

- Dijkstra($G = (V, E, \omega)$):
  - » $\forall\ v \in V$: $d[v] \leftarrow \infty$, $p[v] \leftarrow \emptyset$
  - » $H \leftarrow \{(s, 0, \emptyset)\}$
  - » while(H)
    - – $(v, \delta, \pi) \leftarrow$ RemoveSmallestδ(H)
    - – if($\delta < d[v]$)
      - $d[v] \leftarrow \delta$, $p[v] \leftarrow \pi$
      - $\forall (v, w) \in E$
        - $\eta \leftarrow \delta + \omega(v, w)$
        - if($\eta < d[w]$)
          - Insert(H , (w, η, v))



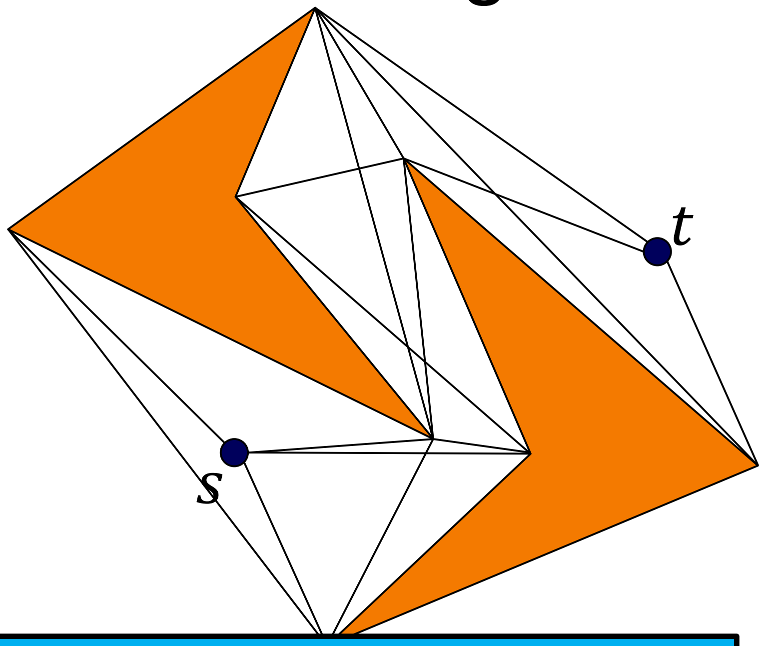
If($d[t] = \infty$): t is unreachable
Else: the path from s to t is $\{s, \ldots, p\big[p[t]\big], p[t], t\}$

Visibility Graph

Naïve Algorithm:

For each pair of nodes, $O(N^2)$, and each polygon edge, $O(N)$, test if the segment between the nodes intersects the edge.

$\Rightarrow$ Complexity: $O(N^3)$



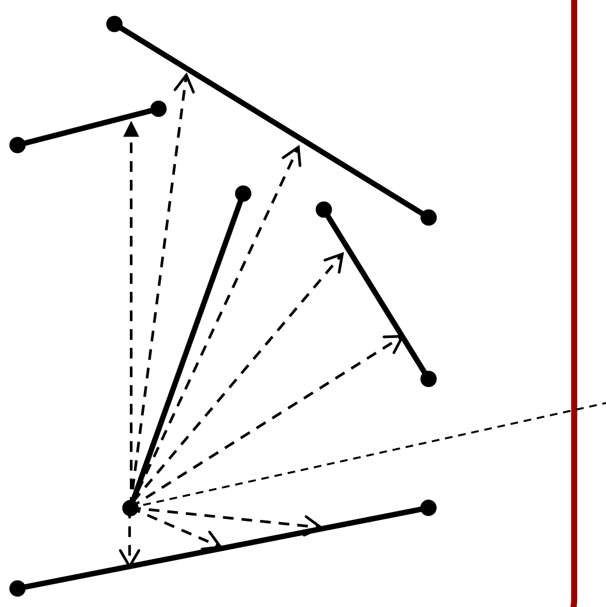
Note that the visibility graph can have $O(N^2)$ edges.

Visibility Graph [Lee 1987]

Given non-intersecting edges $E \subset V \times V$, construct the visibility graph by first constructing the reachability function:

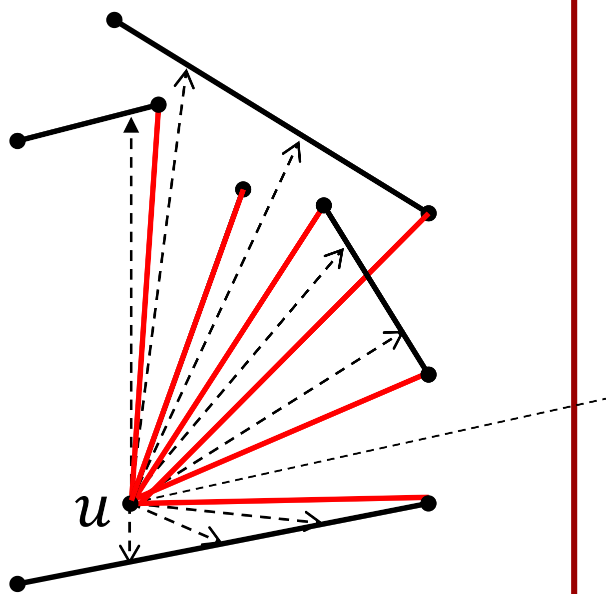
$$R: V \times [-\infty, \infty] \to E$$

This gives the first edge hit when travelling from $v \in V$ along a line to the right with slope $m \in [-\infty, \infty]$.



Visibility Graph [Lee 1987]

Because the edges in E are non-intersecting, the label $R(u,*)$ can only change when the slope aligns with a segment $(u,*) \in V \times V$.

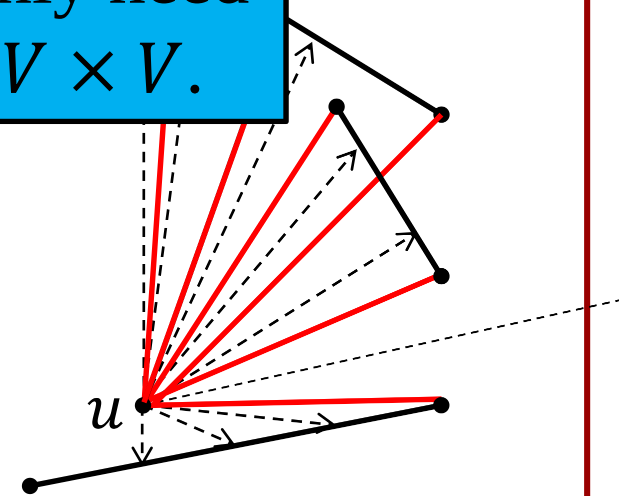


Visibility Graph [Lee 1987]

Because the edges in E are non-intersecting, the label $R(u,*)$ can only change when the slo

Instead of considering the whole continuum $[-\infty,\infty]$, it suffices to consider the $O(N^2)$ slopes generated by the vertex pairs $V\times V$.

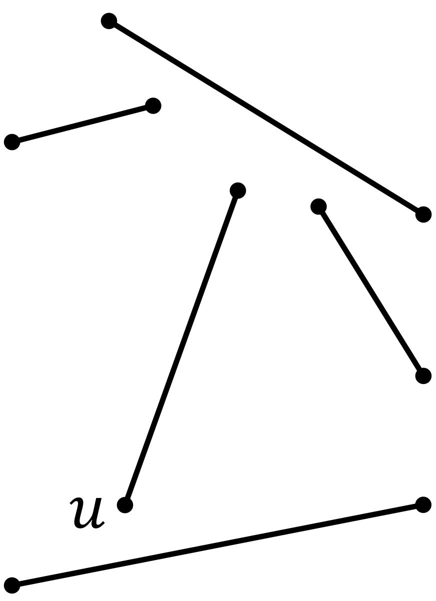
For a given vertex $u\in V$, we only need to consider the pairs $(u,v)\in V\times V$.



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

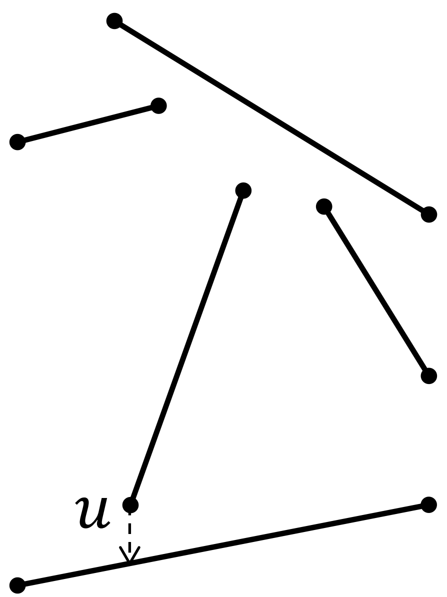
- $S \leftarrow$SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - $R(u, \angle uv) \leftarrow R(u, \angle uv - \epsilon)$
    - Output uv
  - » else if($w = \emptyset$ or $|uv| < |uw|$)
    - $R(u, \angle uv) \leftarrow e(v)$
    - Output uv
  - » else if($|uv| > |uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

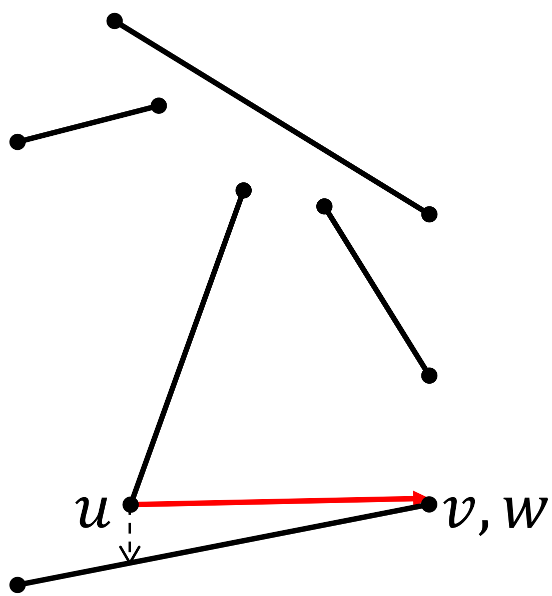
- $S \leftarrow$SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(u, \angle uv - \epsilon)$
    - – Output uv
  - » else if($w = \emptyset$ or $|uv| < |uw|$)
    - – $R(u, \angle uv) \leftarrow e(v)$
    - – Output uv
  - » else if($|uv| > |uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G=(V,E)$):

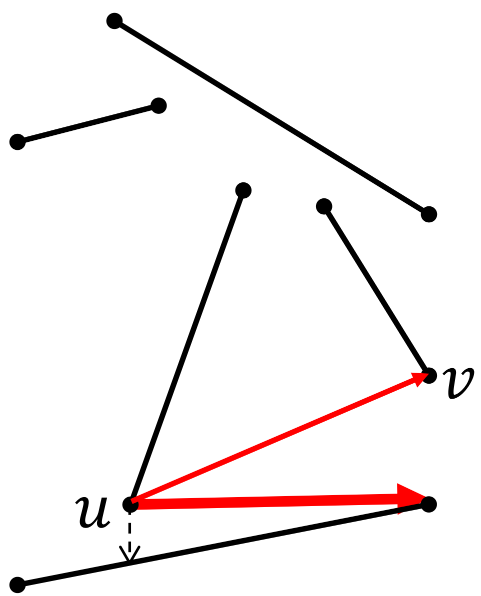
- $S \leftarrow$SortBySlope($V\times V$)
- Compute $R(*,-\infty)$
- For $(u,v)\in S$ (w/ $u_x<v_x$)
  - » $w\leftarrow R(u,\angle uv-\epsilon)\cap\overrightarrow{uv}$
  - » if($(u,v)\in E$): Output uv
  - » else if($v=w$)
    - – $R(u,\angle uv)\leftarrow R(v,\angle uv-\epsilon)$
    - – Output uv
  - » else if($w=\emptyset$ or $|uv|<|uw|$)
    - – $R(u,\angle uv)\leftarrow e(v)$
    - – Output uv
  - » else if($|uv|>|uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

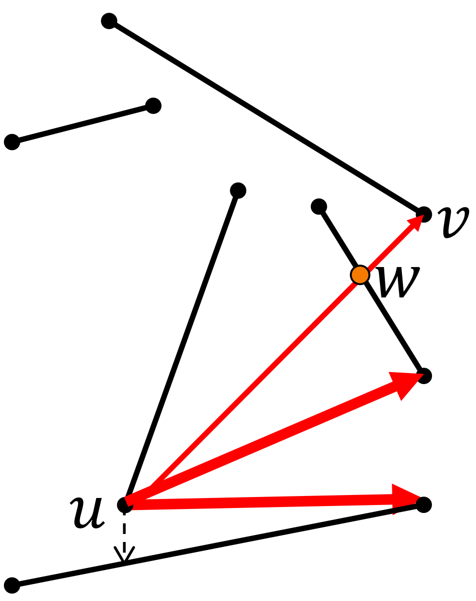
- $S \leftarrow$SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(v, \angle uv - \epsilon)$
    - – Output uv
  - » <u>else if($w = \emptyset$ or $|uv| < |uw|$)</u>
    - – <u>$R(u, \angle uv) \leftarrow e(v)$</u>
    - – <u>Output uv</u>
  - » else if($|uv| > |uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

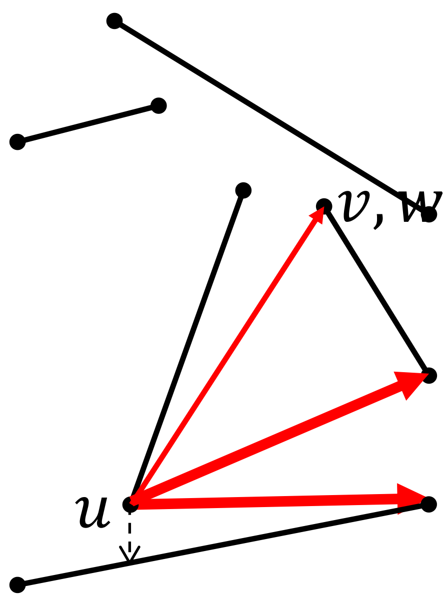
- $S \leftarrow$SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(v, \angle uv - \epsilon)$
    - – Output uv
  - » else if($w = \emptyset$ or $|uv| < |uw|$)
    - – $R(u, \angle uv) \leftarrow e(v)$
    - – Output uv
  - » <u>else if($|uv| > |uw|$): Do nothing</u>



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

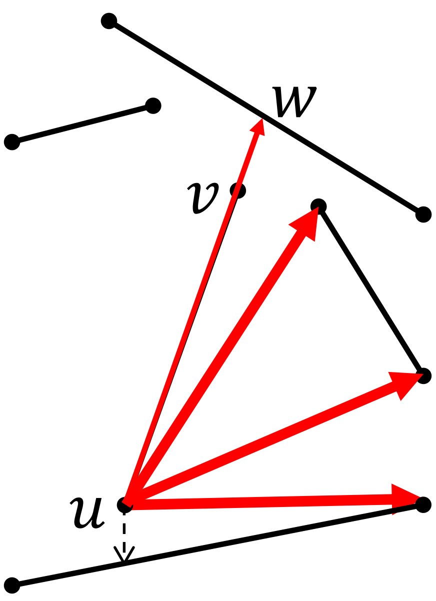
- $S \leftarrow$SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(v, \angle uv - \epsilon)$
    - – Output uv
  - » else if($w = \emptyset$ or $|uv| < |uw|$)
    - – $R(u, \angle uv) \leftarrow e(v)$
    - – Output uv
  - » else if($|uv| > |uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

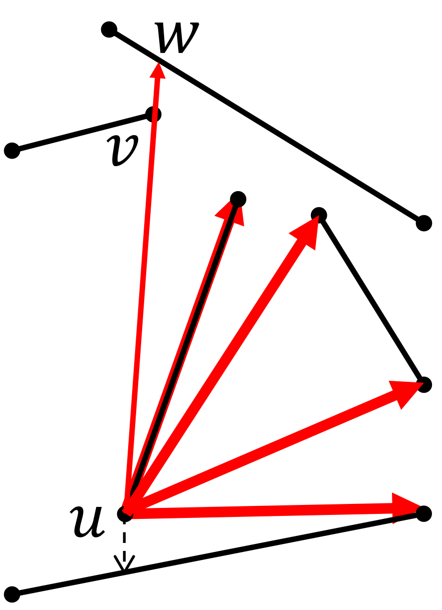
- $S \leftarrow$SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(v, \angle uv - \epsilon)$
    - – Output uv
  - » else if($w = \emptyset$ or $|uv| < |uw|$)
    - – $R(u, \angle uv) \leftarrow e(v)$
    - – Output uv
  - » else if($|uv| > |uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

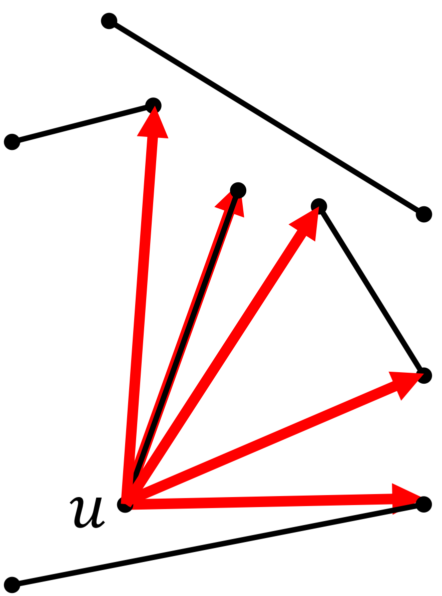
- $S \leftarrow$SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(v, \angle uv - \epsilon)$
    - – Output uv
  - » <u>else if($w = \emptyset$ or $|uv| < |uw|$)</u>
    - – <u>$R(u, \angle uv) \leftarrow e(v)$</u>
    - – <u>Output uv</u>
  - » else if($|uv| > |uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

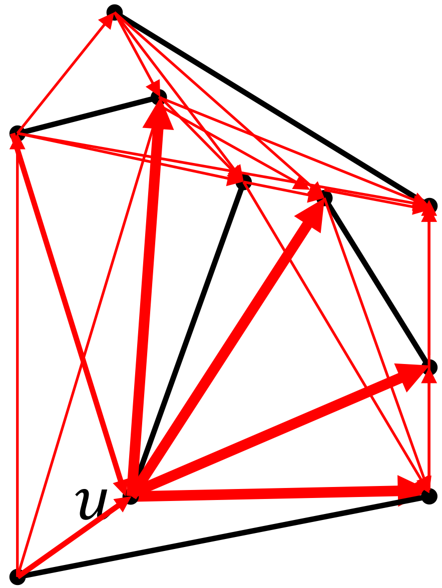
- S ←SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(v, \angle uv - \epsilon)$
    - – Output uv
  - » else if($w = \emptyset$ or $|uv| < |uw|$)
    - – $R(u, \angle uv) \leftarrow e(v)$
    - – Output uv
  - » else if($|uv| > |uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

- S ←SortBySlope($V \times V$)
- Compute $R(*, -\infty)$
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(v, \angle uv - \epsilon)$
    - – Output uv
  - » else if($w = \emptyset$ or $|uv| < |uw|$)
    - – $R(u, \angle uv) \leftarrow e(v)$
    - – Output uv
  - » else if($|uv| > |uw|$): Do nothing

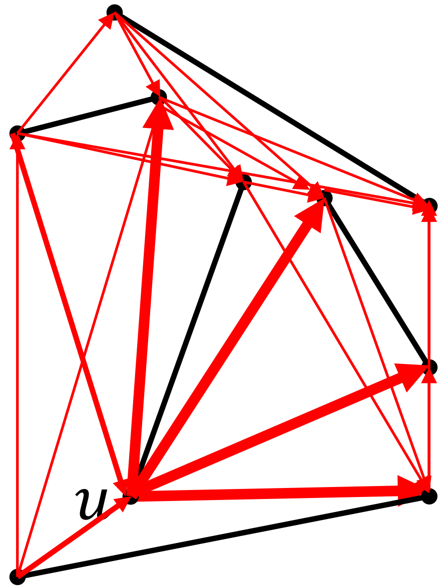


Visibility Graph [Lee 1987]

Because segments are sorted, when we process slope $\angle uv$, we can assume that $R(*, \angle uv - \epsilon)$ has been set correctly

Visibil

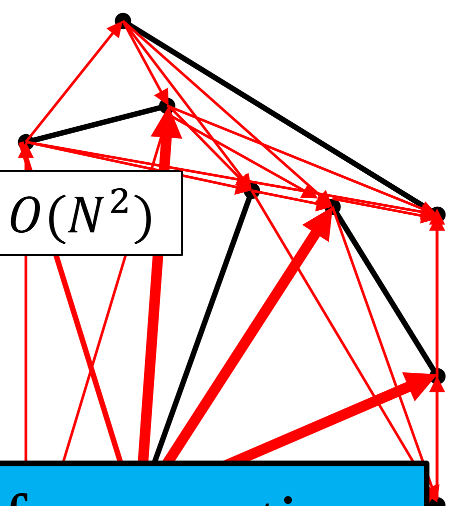
- $S \leftarrow$
- Con
- For $(u, v) \in S$ (w/ $u_x < v_x$)
  - » $w \leftarrow \boxed{R(u, \angle uv - \epsilon)} \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - $R(u, \angle uv) \leftarrow \boxed{R(v, \angle uv - \epsilon)}$
    - Output uv
  - » else if($w = \emptyset$ or $|uv| < |uw|$)
    - $R(u, \angle uv) \leftarrow e(v)$
    - Output uv
  - » else if($|uv| > |uw|$): Do nothing



Visibility Graph [Lee 1987]

VisbilityGraph($G = (V, E)$):

- $S \leftarrow$ SortBySlope($V \times V$) — $O(N^2 \log N)$
- Compute $R(*, -\infty)$ — $O(N^2)$
- For $(u, v) \in S$ (w/ $u_x < v_x$) — $O(N^2)$
  - » $w \leftarrow R(u, \angle uv - \epsilon) \cap \overrightarrow{uv}$
  - » if($(u, v) \in E$): Output uv
  - » else if($v = w$)
    - – $R(u, \angle uv) \leftarrow R(v, \angle uv - \epsilon)$
    - – Output uv
  - » else if($|uv| < |uw|$)
    - – $R(u, \angle uv) \leftarrow e(v)$
  - » else if(|



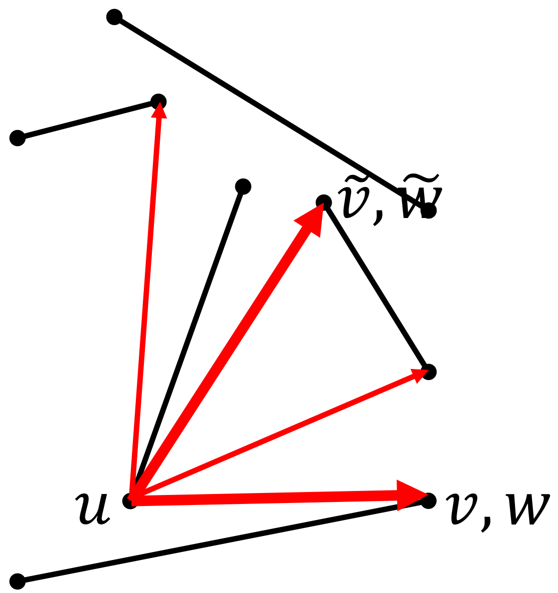
The $O(N^2 \log N)$ complexity comes from sorting.

Can we sort more quickly?

Visibility Graph [Welzl 1985]

To compute visibility graph, we didn't require that all edges in $S = V \times V$ sorted. Rather:

- Each subset of segments with the same left-most vertex had to be sorted.
- If $(v = w)$ then the reach of v with slope $\angle uv - \epsilon$ had to have been already set.



Visibility Graph [Welzl 1985]

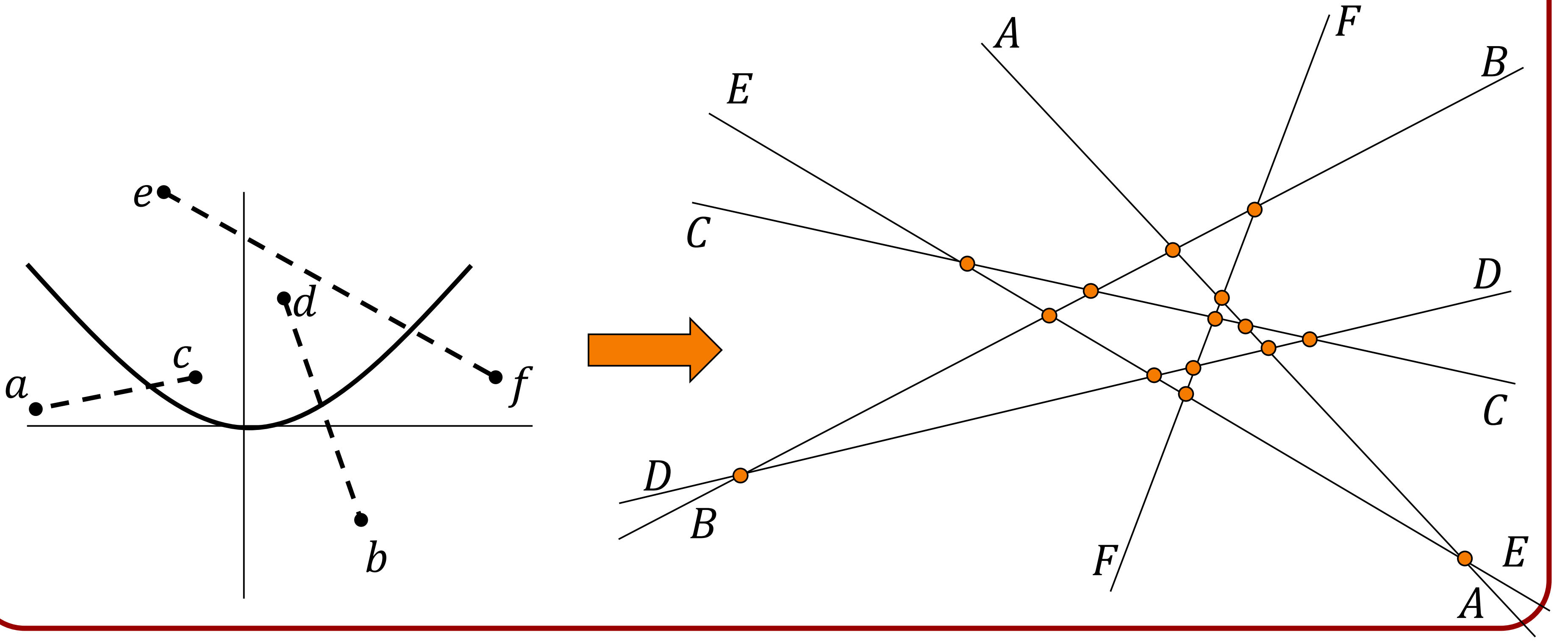
Recall:

- An arrangement of N lines can be computed in $O(N^2)$ time.
- Given a line $L = \{(x, y) | y = mx + b\} \subset \mathbb{R}^2$, it's dual is the point:
$$D(L) = \left(\frac{m}{2}, -b\right)$$
In particular, the x-coordinate of the dual of a line is (half) the line's slope.

Visibility Graph [Welzl 1985]

Approach:

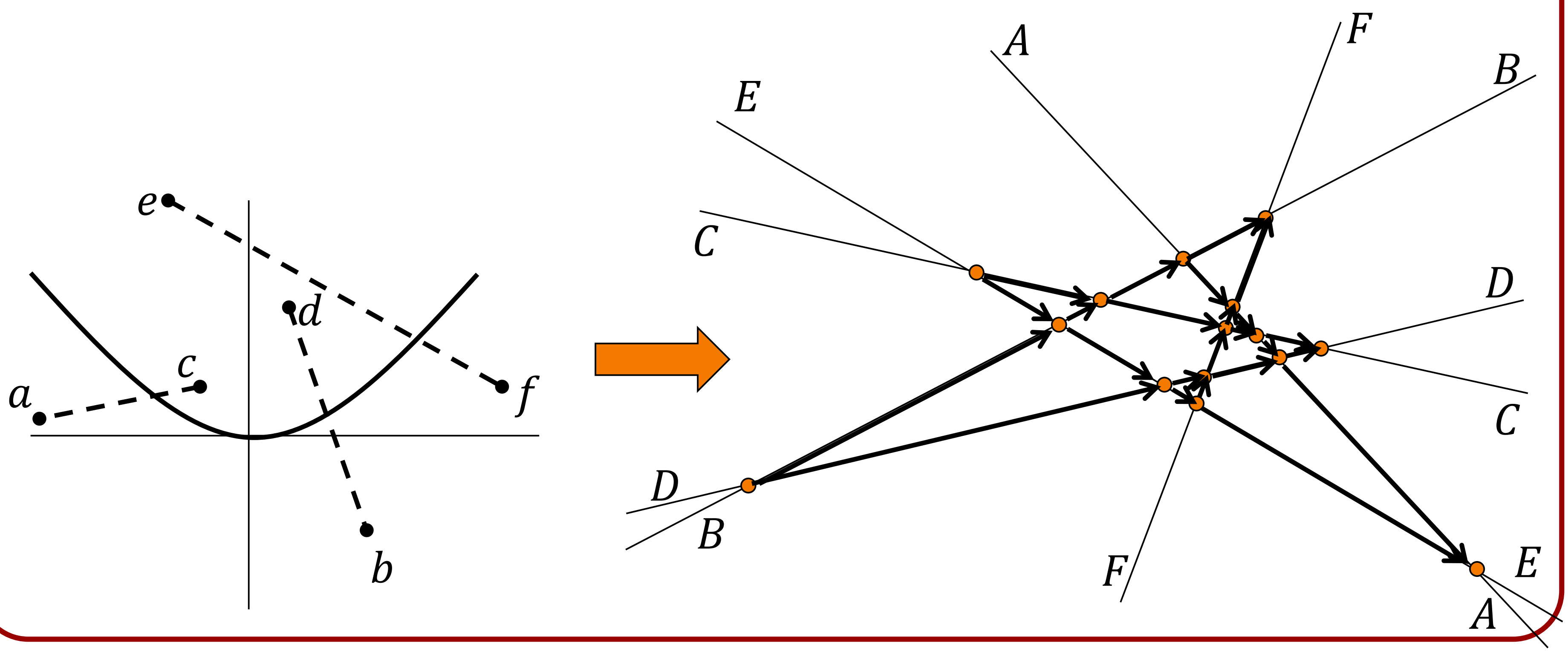
- Compute the line arrangement dual to V.



Visibility Graph [Welzl 1985]

Approach:

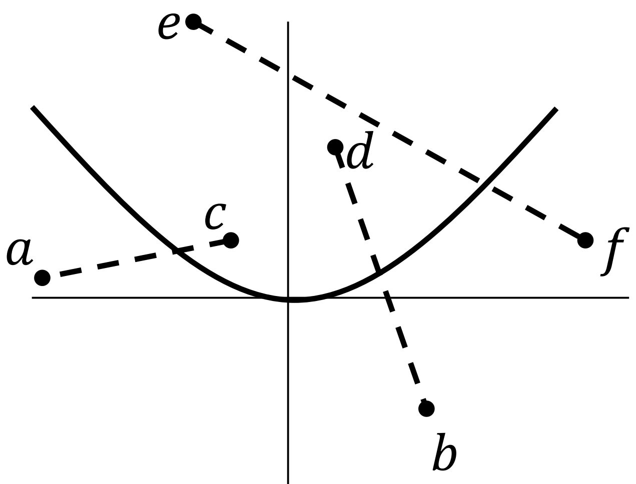
- Define a directed graph on the dual:
  - $v \rightarrow w$ if v and w are adjacent and $v_x < w_x$.

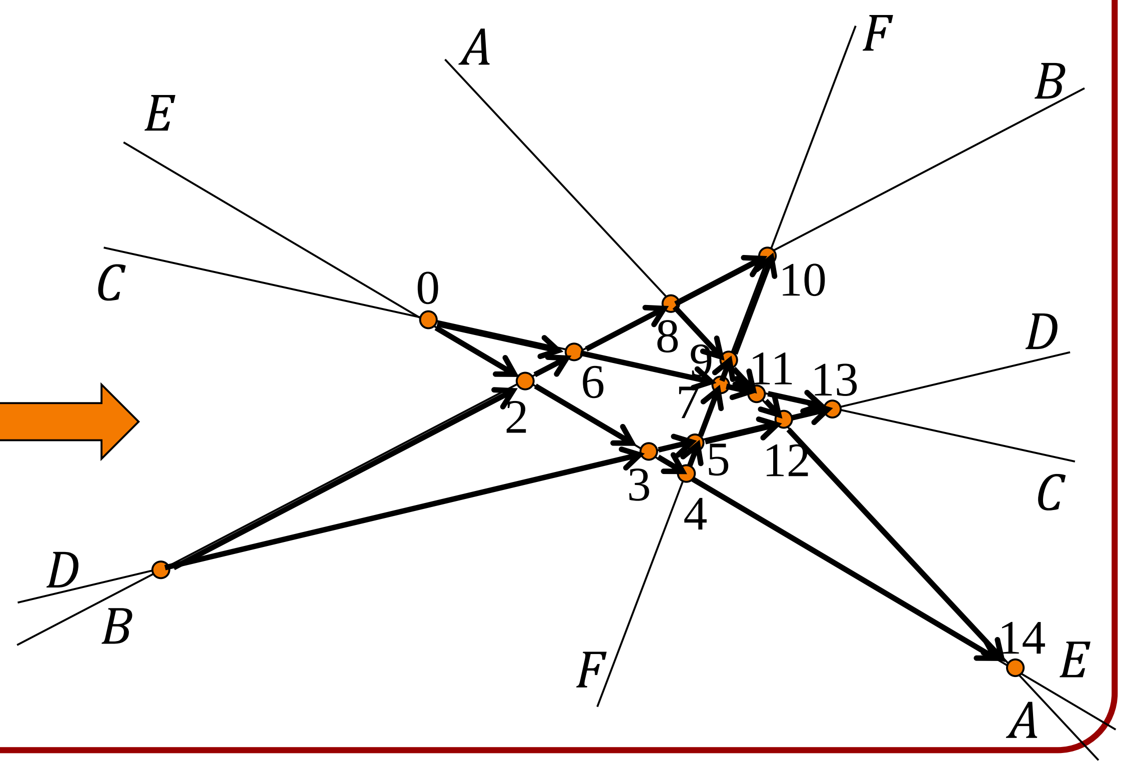


Visibility Graph [Welzl 1985]

Approach:

- Perform a topological sort on the directed graph:
  - Assign integer indices to the nodes s.t. $u \to v \Rightarrow u < v$.



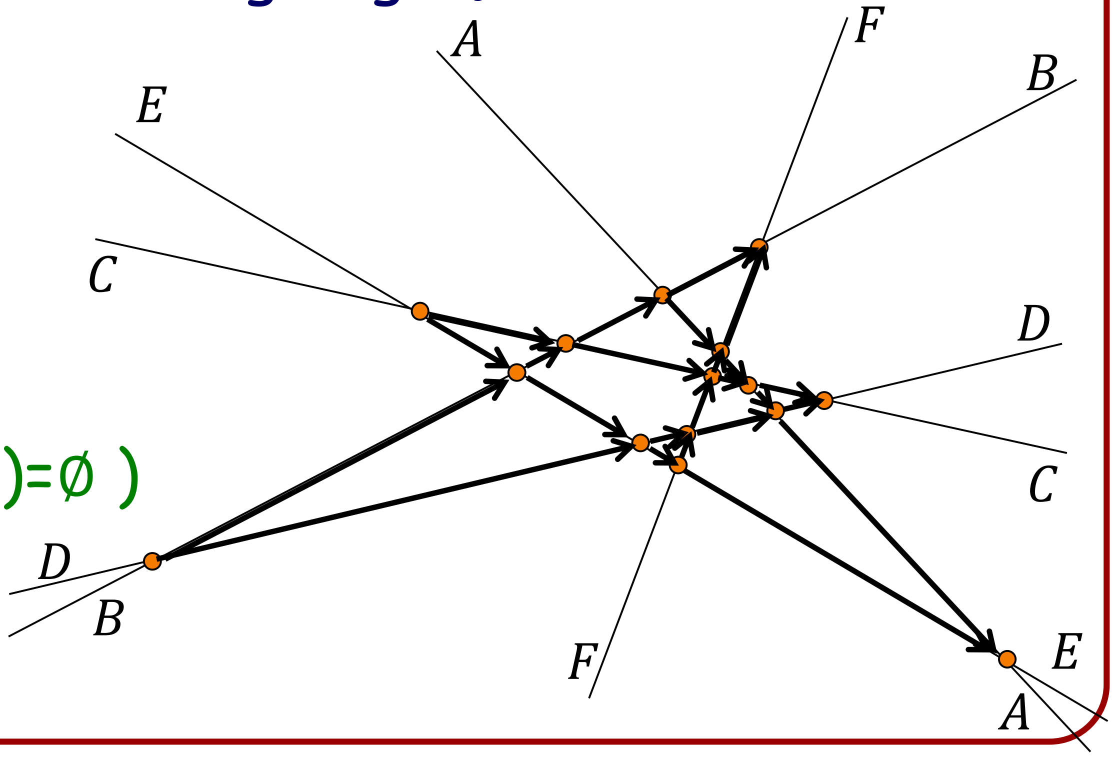


Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

TopSort($G=(V,E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v,w) \in E$
    - – $E \leftarrow E - \{(v,w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



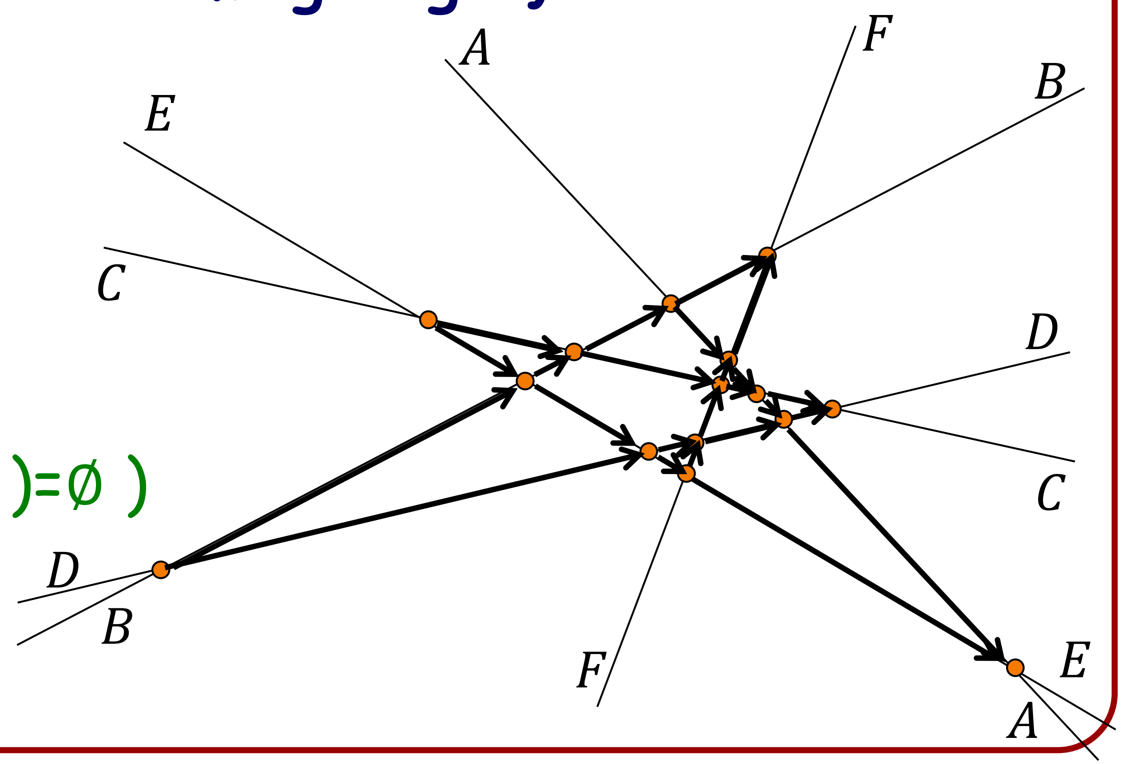
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{BD, CE\}$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ **{vertices with no incoming edges}**
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)

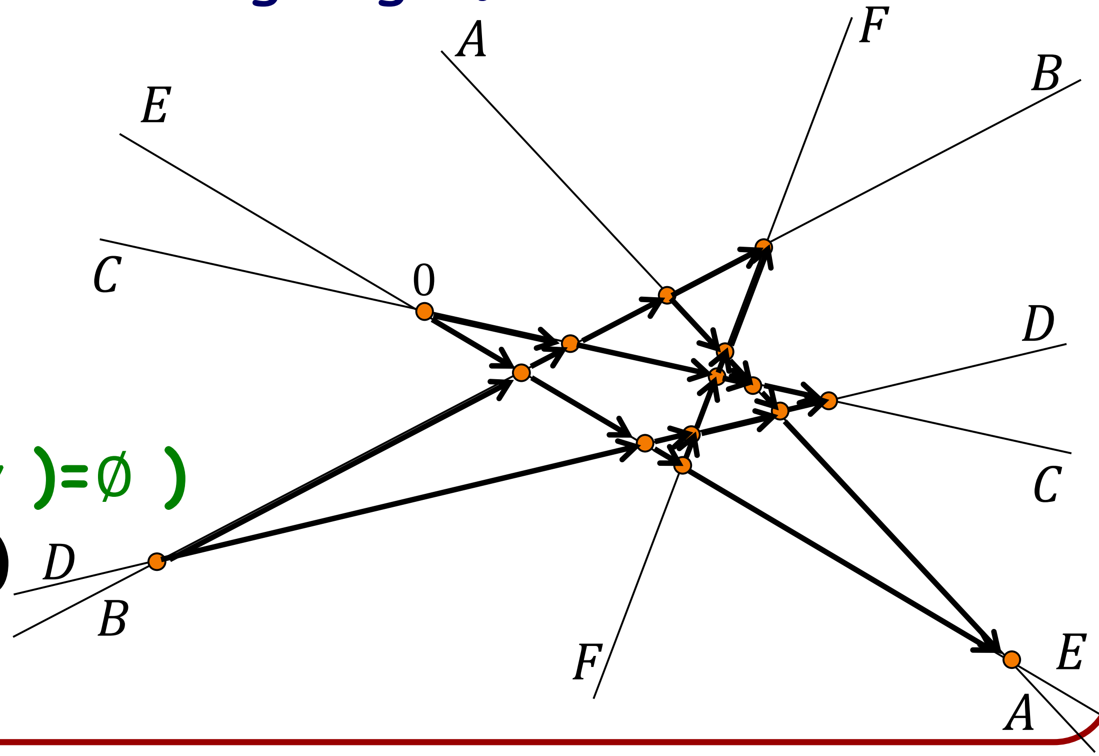


Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{BD\}$
$v = CE$

TopSort($G = (V, E)$):
- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



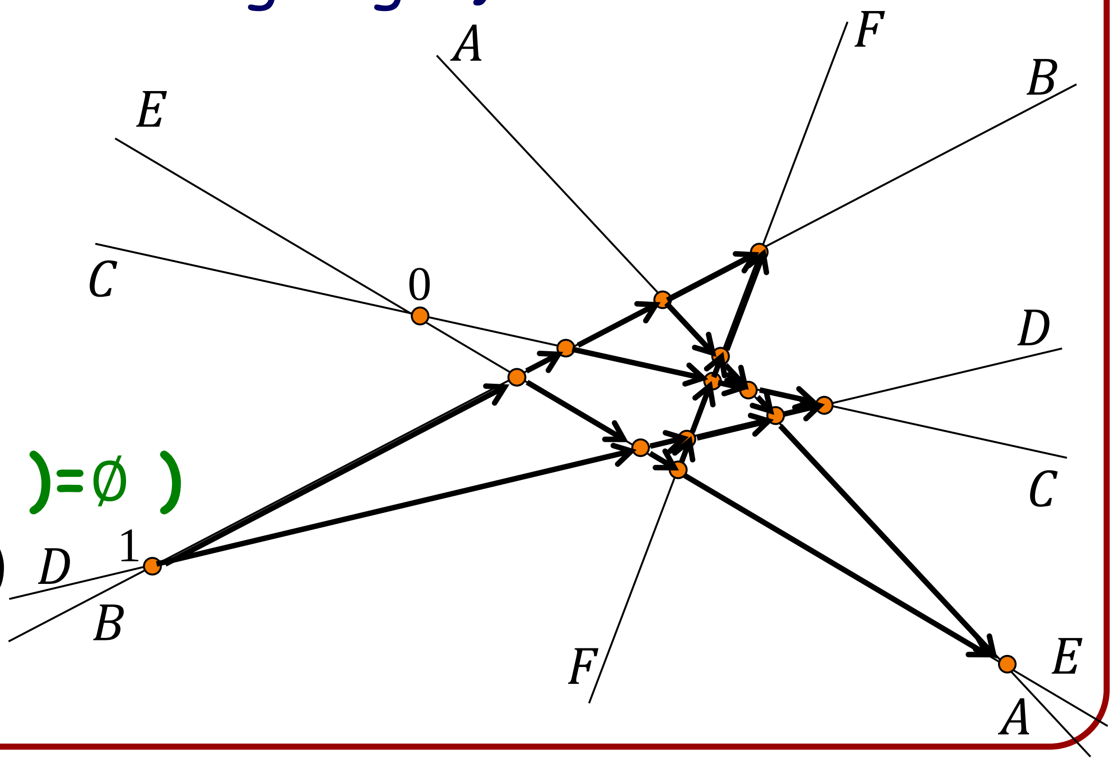
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{BE\}$
$v = BD$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



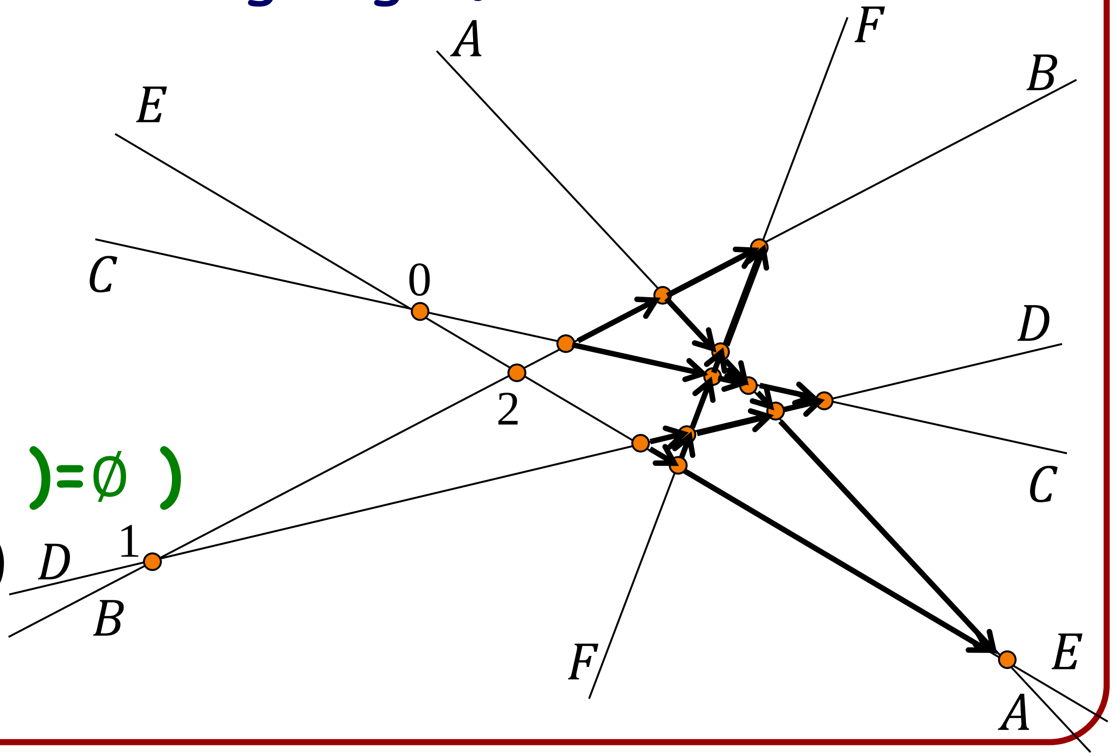
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{BC, DE\}$
$v = BE$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)

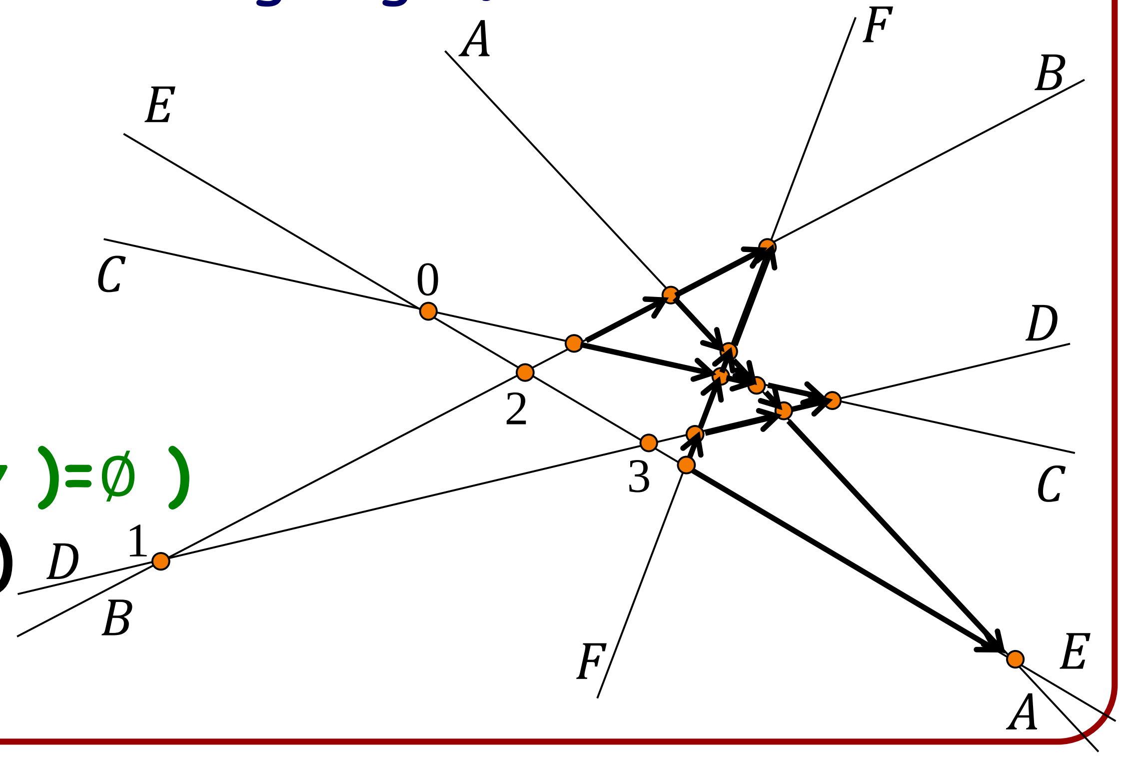


Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

TopSort($G = (V, E)$):
- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)

$S = \{BC, EF\}$
$v = DE$



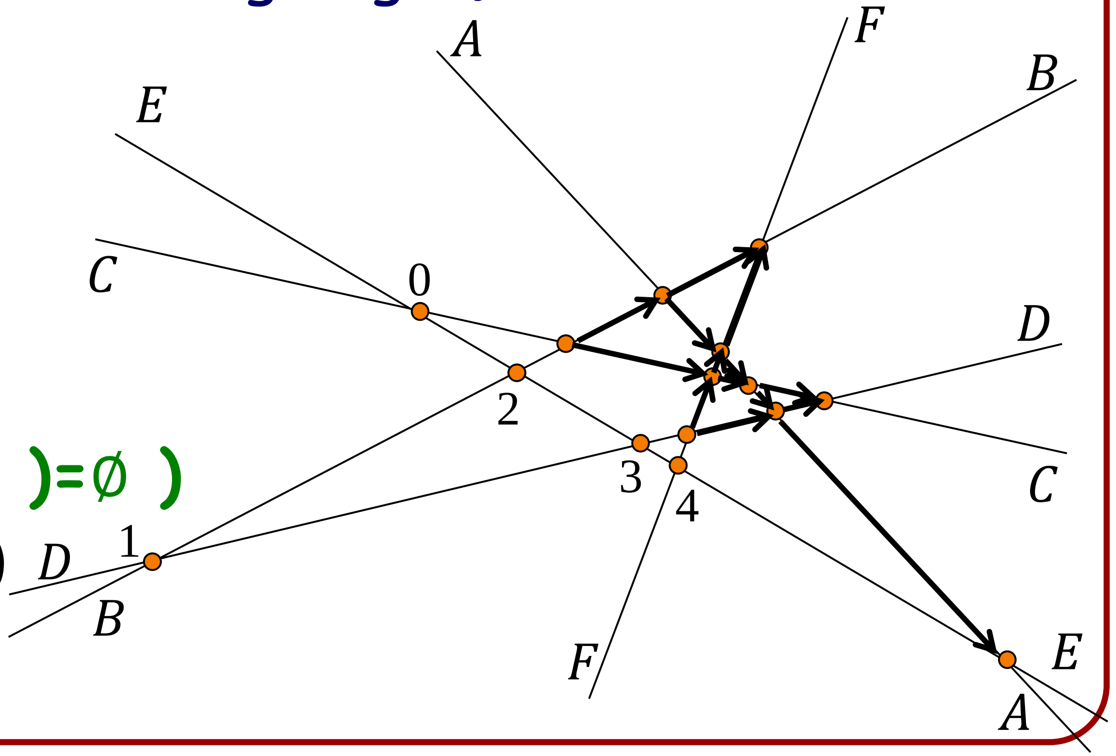
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{BC, DF\}$
$v = EF$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)

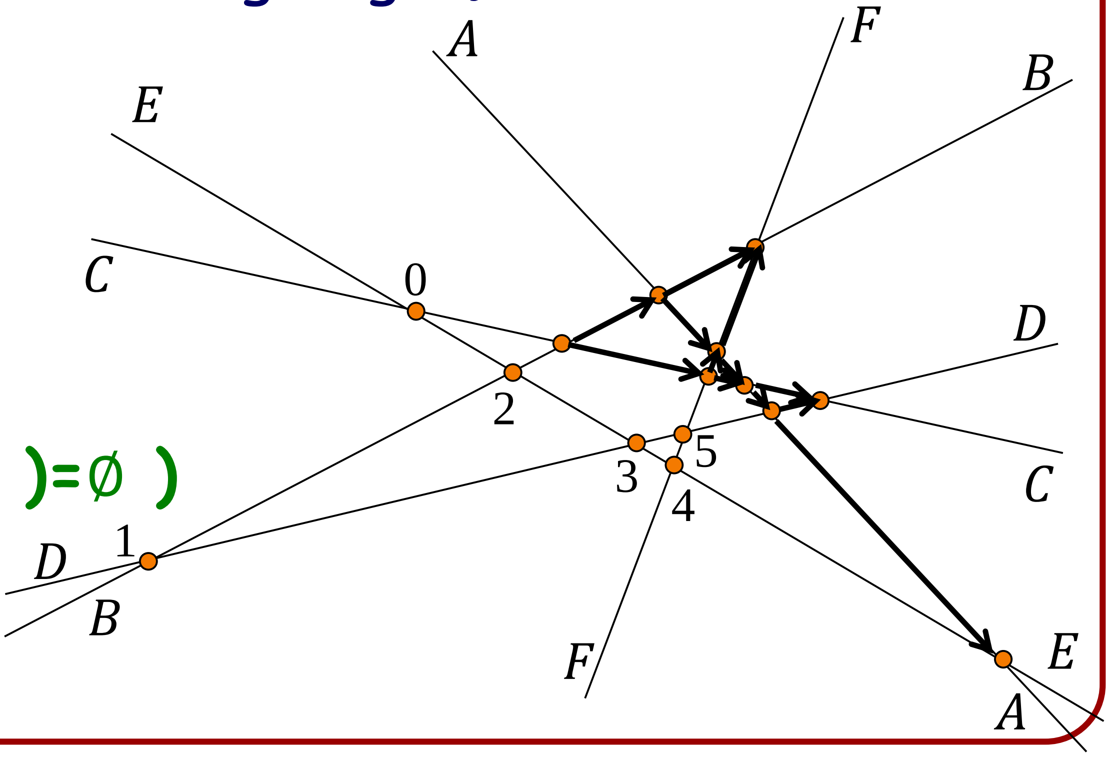


Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{BC\}$
$v = DF$

TopSort($G = (V, E)$):
- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=$\emptyset$)
      - • Push(S , w)



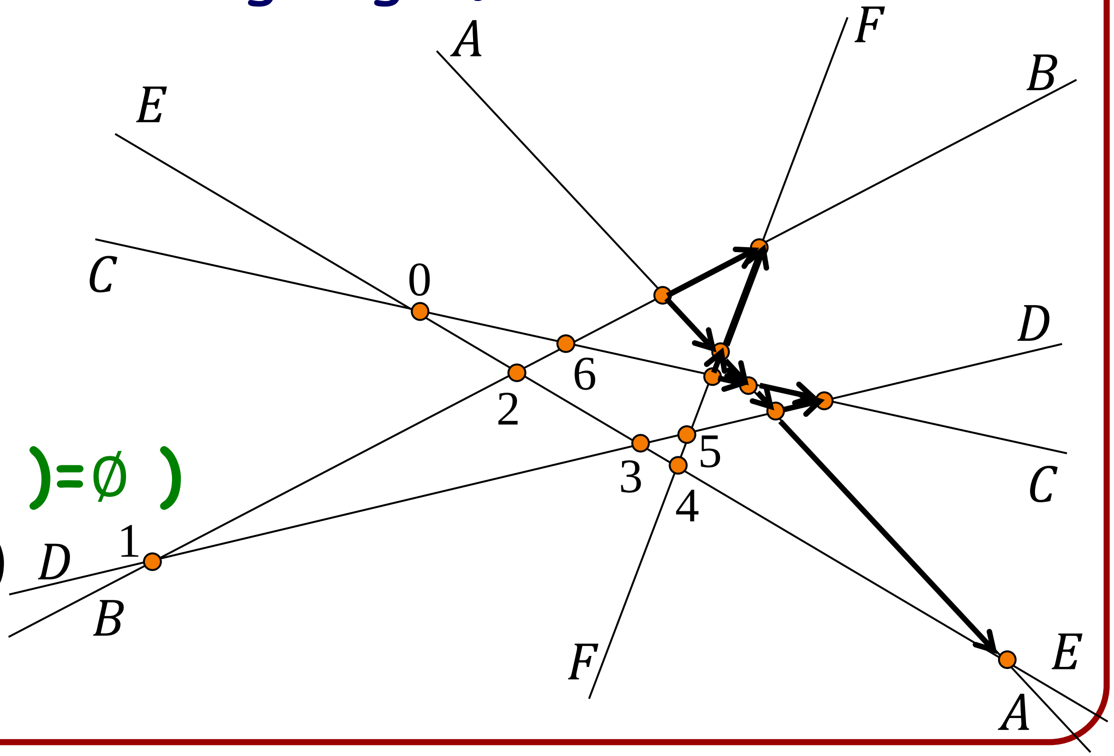
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{AB, CF\}$
$v = BC$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



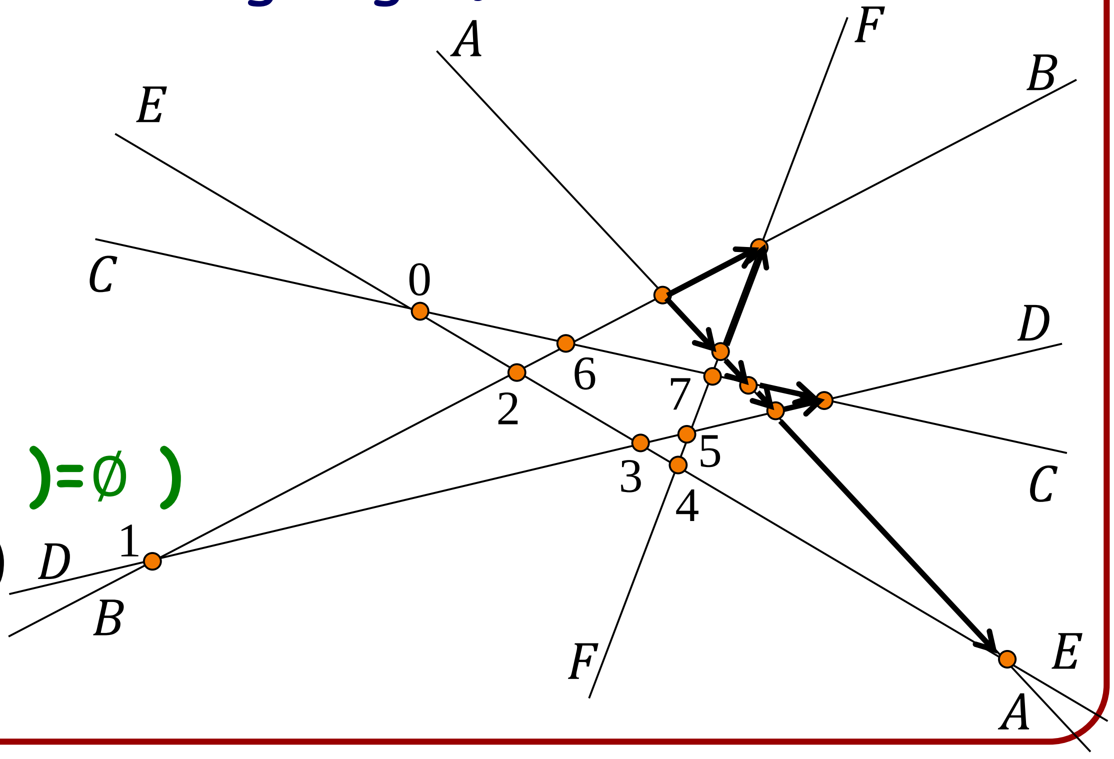
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{AB\}$
$v = CF$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)

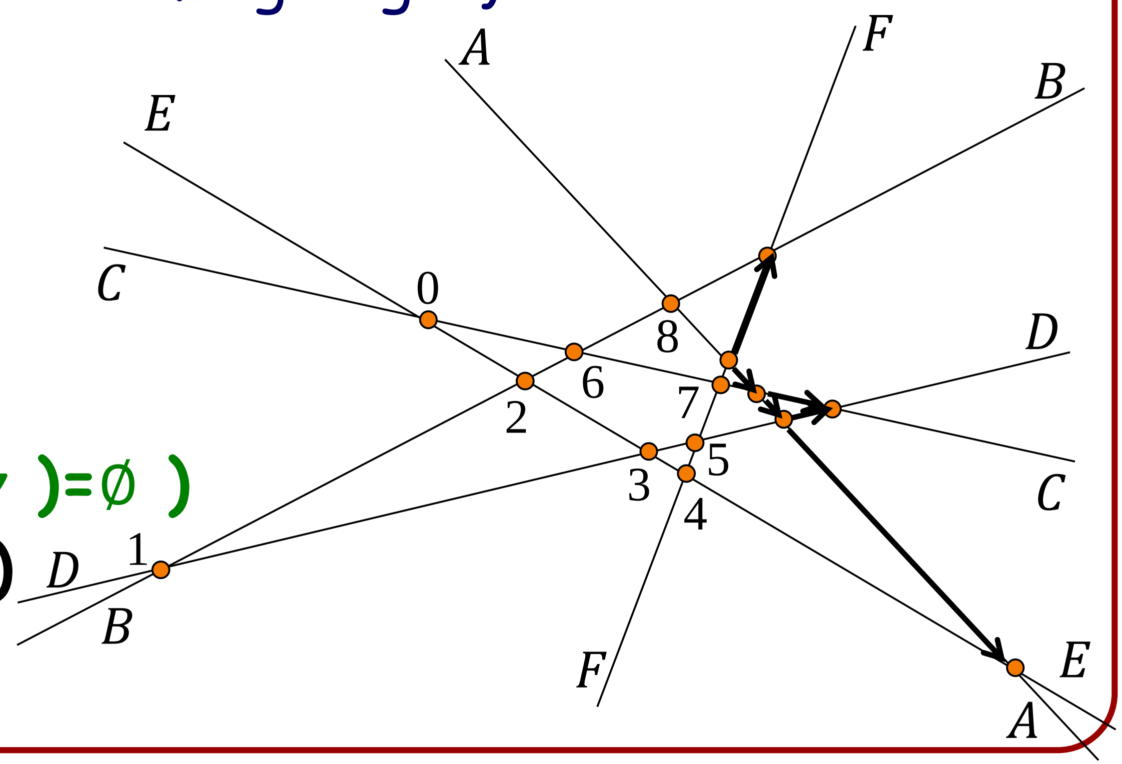


Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{AF\}$
$v = AB$

TopSort($G = (V, E)$):
- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



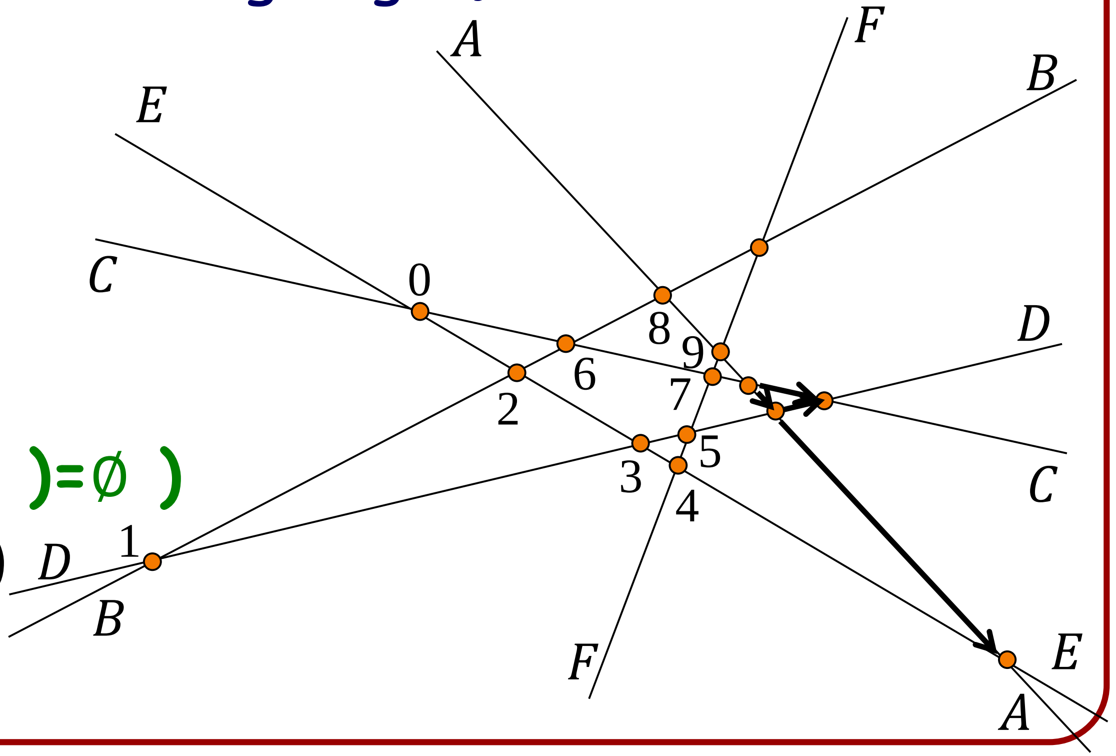
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{AC, BF\}$
$v = AF$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



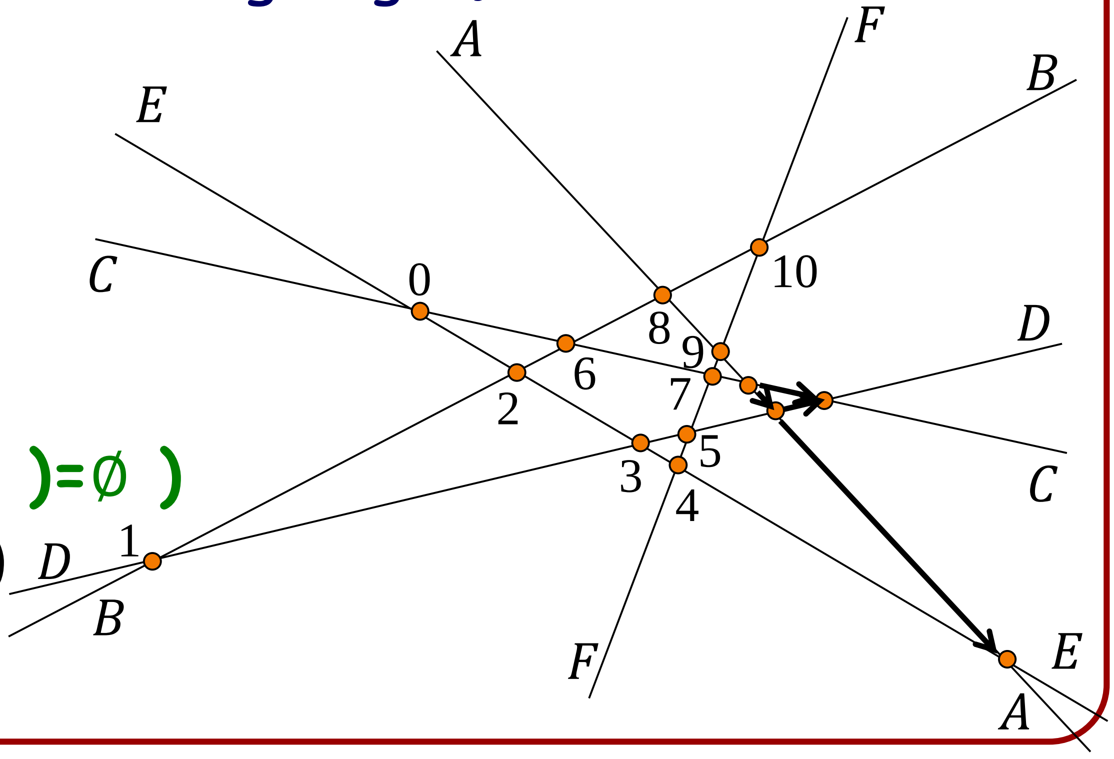
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{AC\}$
$v = BF$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=$\emptyset$)
      - • Push(S , w)



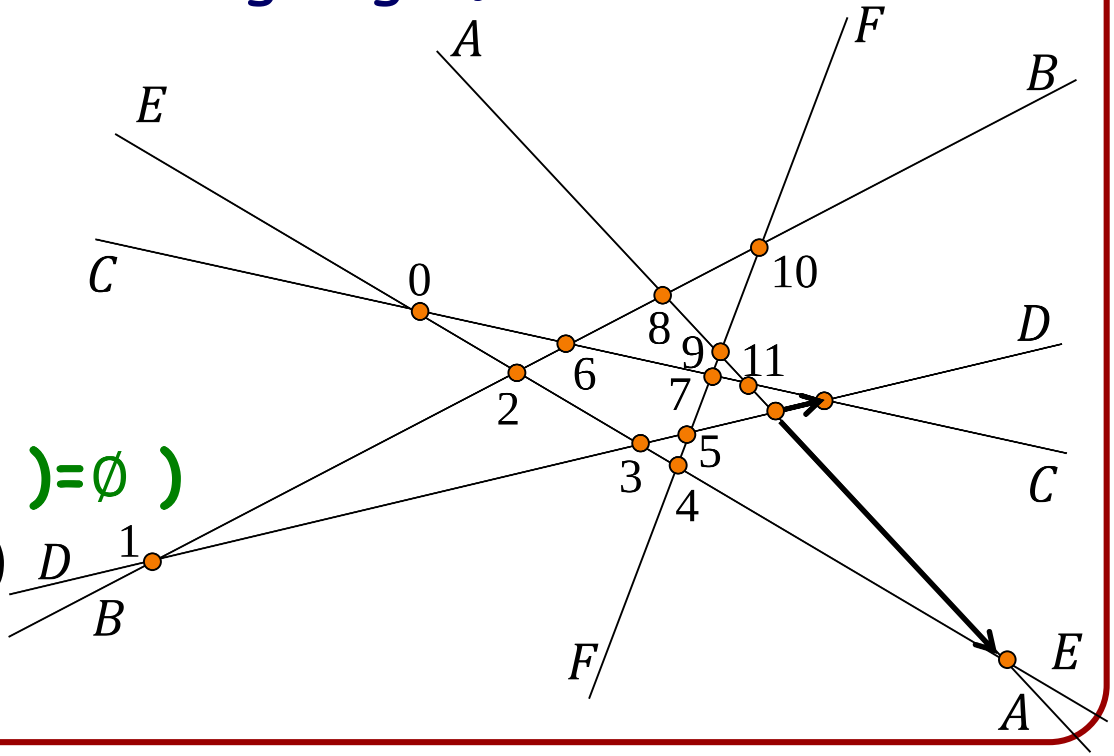
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{AD\}$
$v = AC$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c{+}{+}$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



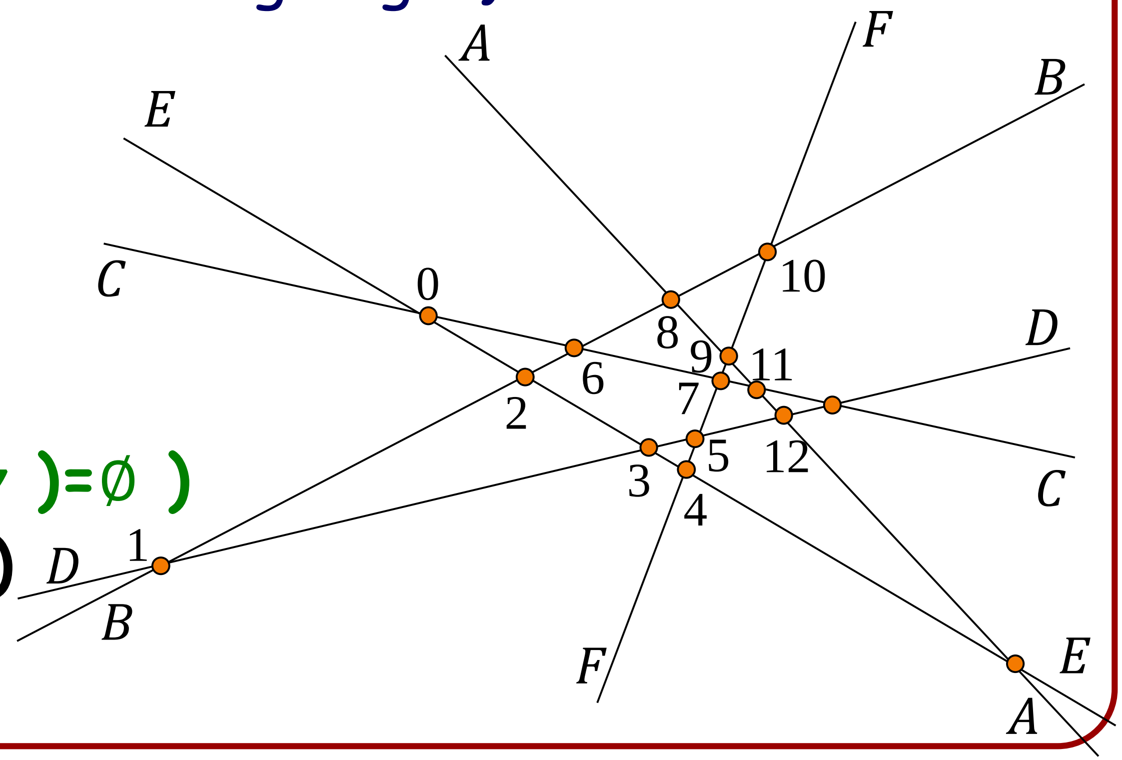
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{AE, CD\}$
$v = AD$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



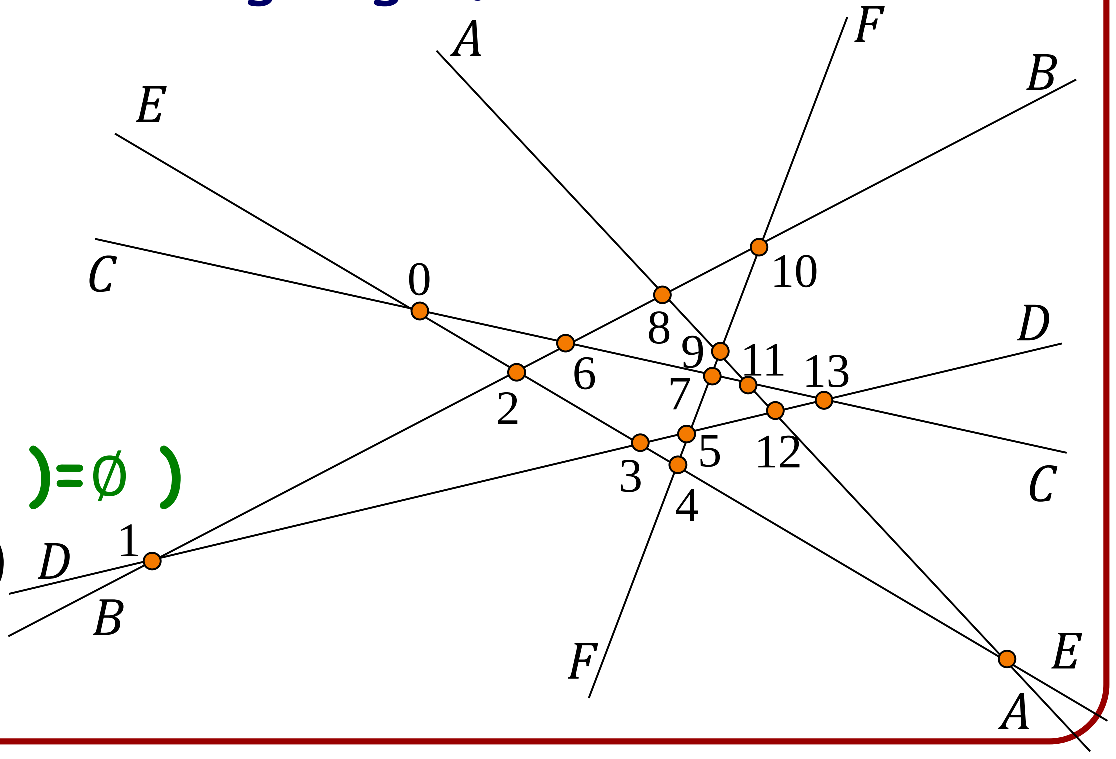
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{AE\}$
$v = CD$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)



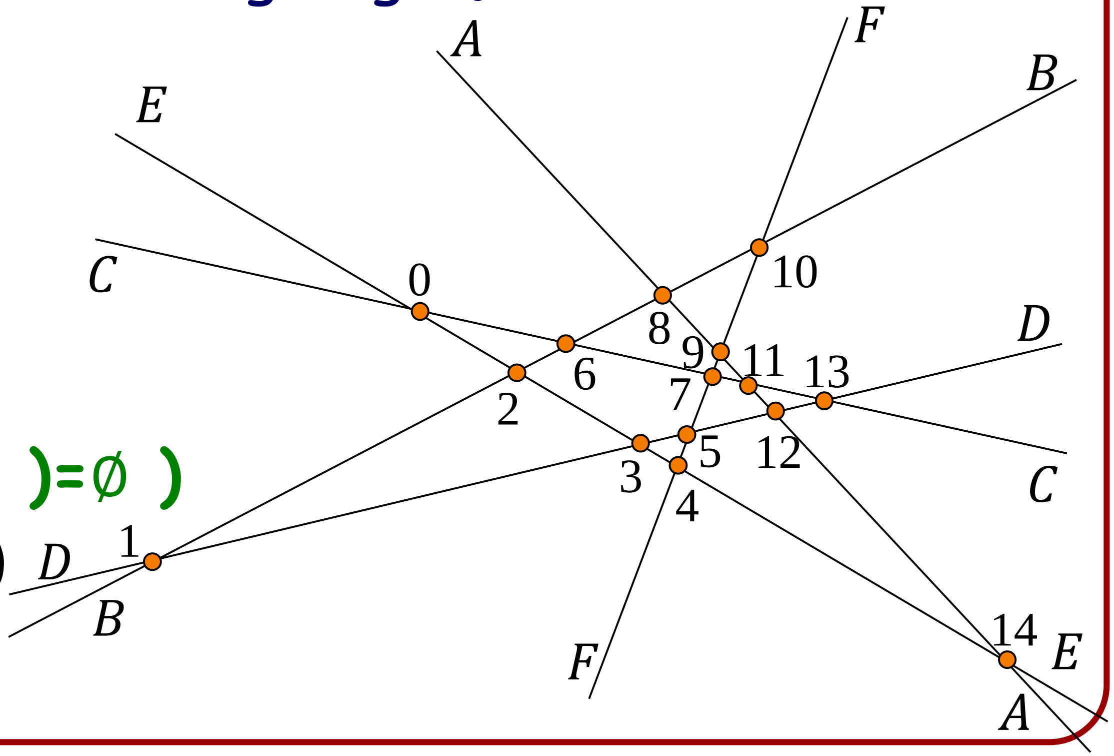
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

$S = \{\}$
$v = AE$

TopSort($G = (V, E)$):

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=$\emptyset$)
      - • Push(S , w)



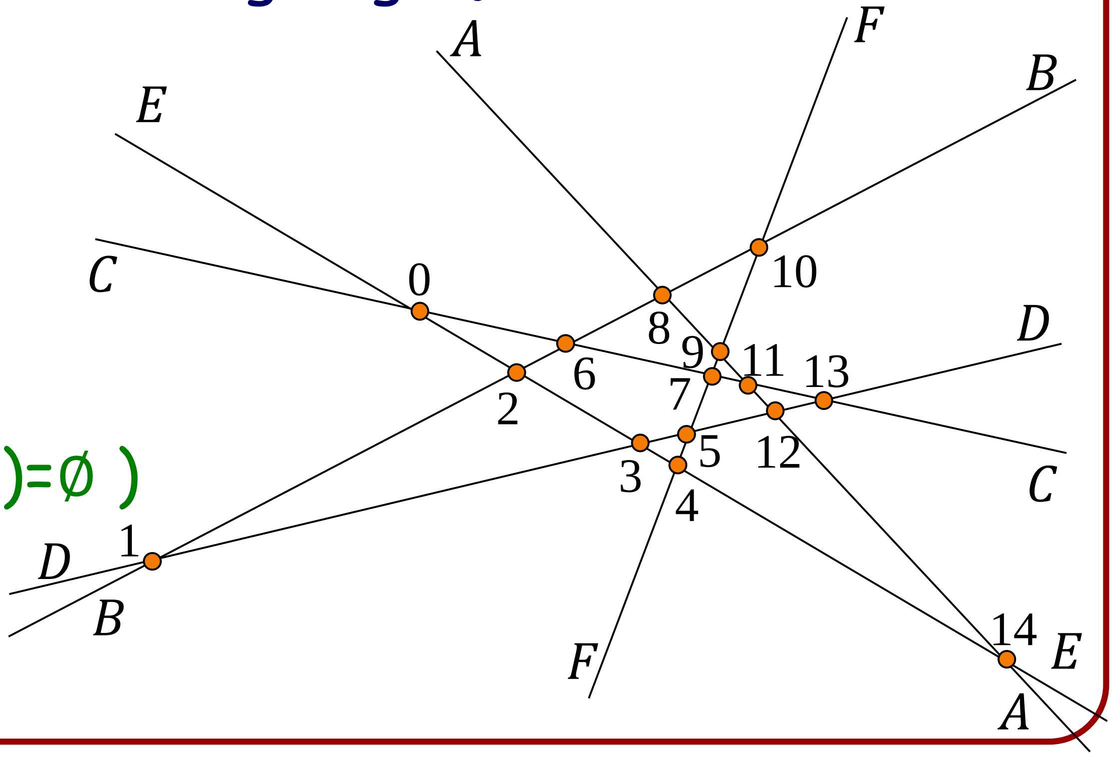
Visibility Graph [Welzl 1985]

Topological Sort [Kahn 1962]:

TopSort($G = (V, E)$):

Complexity: $O(|V| + |E|)$

- $c \leftarrow 0$
- $S \leftarrow$ {vertices with no incoming edges}
- while($S \neq \emptyset$)
  - » $v \leftarrow$ Pop(S)
  - » $s(v) \leftarrow c++$
  - » $\forall\ (v, w) \in E$
    - – $E \leftarrow E - \{(v, w)\}$
    - – if(Incoming(w)=∅)
      - • Push(S , w)

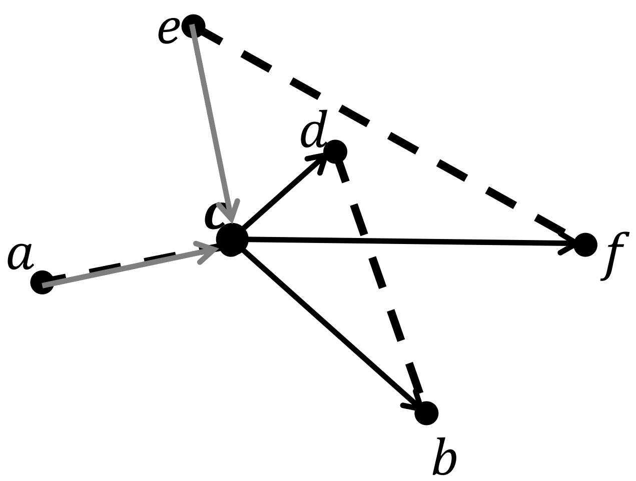


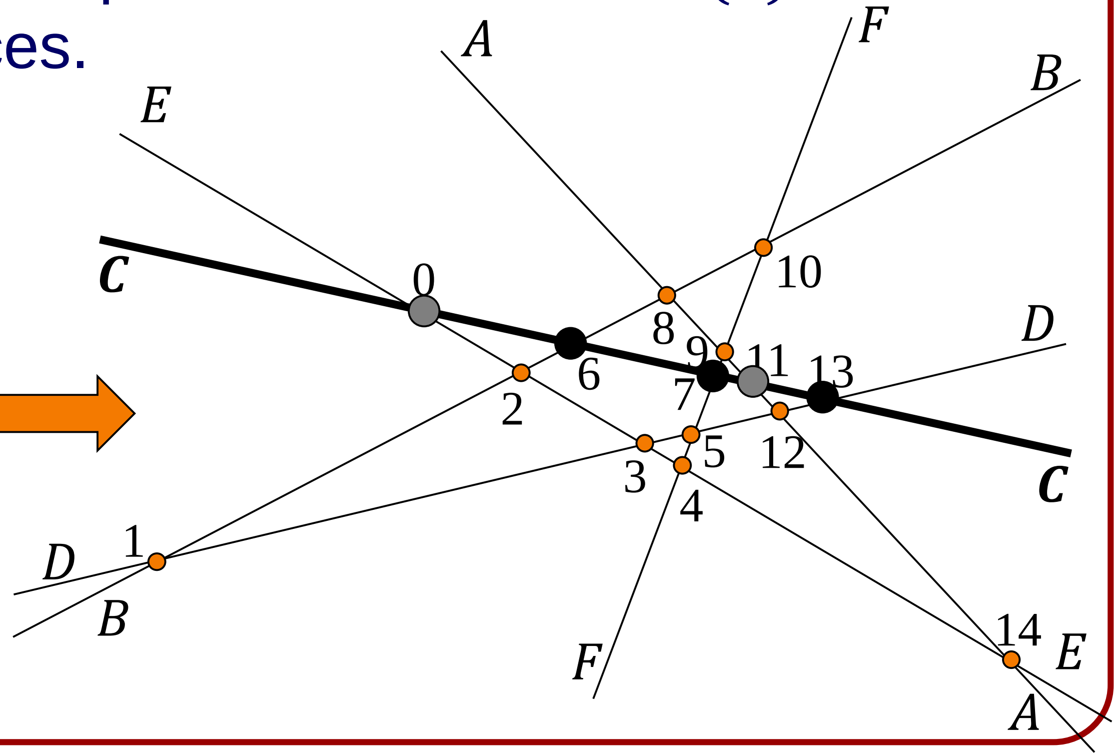
Visibility Graph [Welzl 1985]

<u>Note</u>:

In the dual arrangement:

- A line L corresponds to a vertex $D(L)$ of the input.
- Vertices on L corresponds to lines from $D(L)$ to the other input vertices.

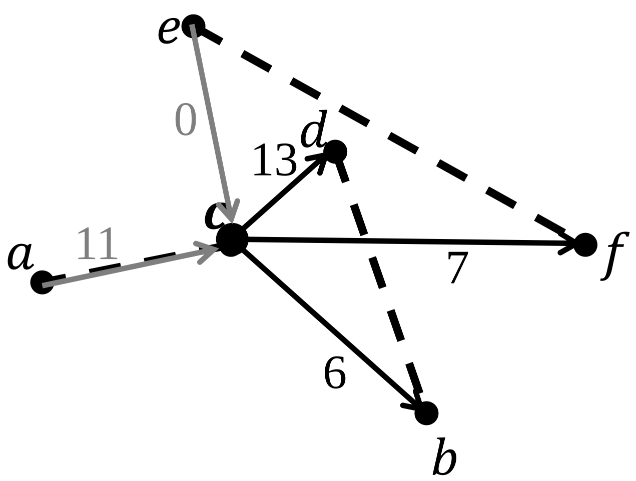


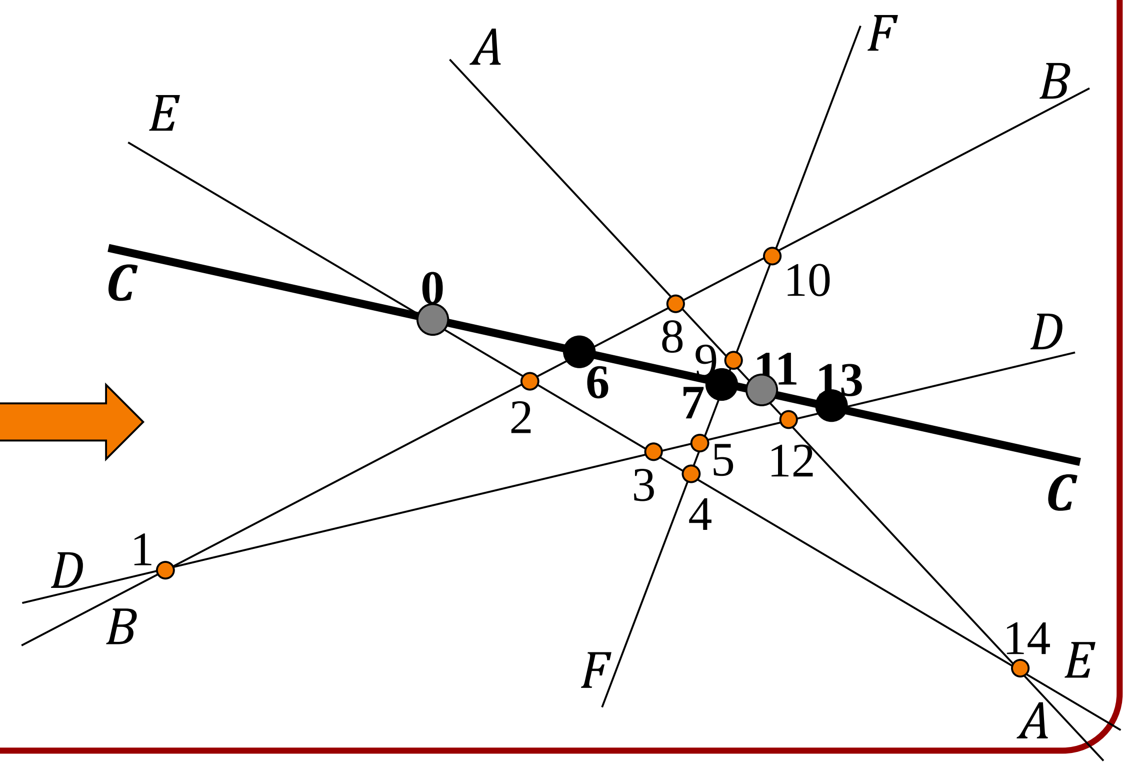


Visibility Graph [Welzl 1985]

For a line in the dual arrangement, the index of vertices on that line, left-to-right, increases.

⇒ The corresponding line segments through the primal vertex have monotonically increasing slope.

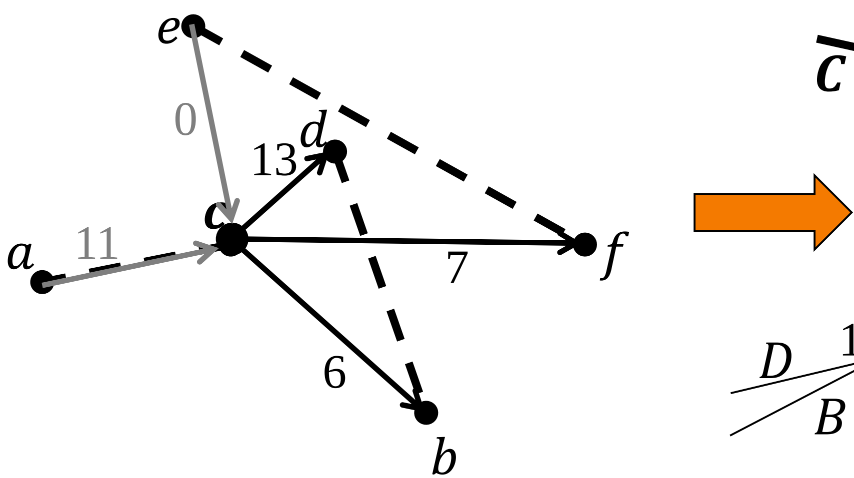


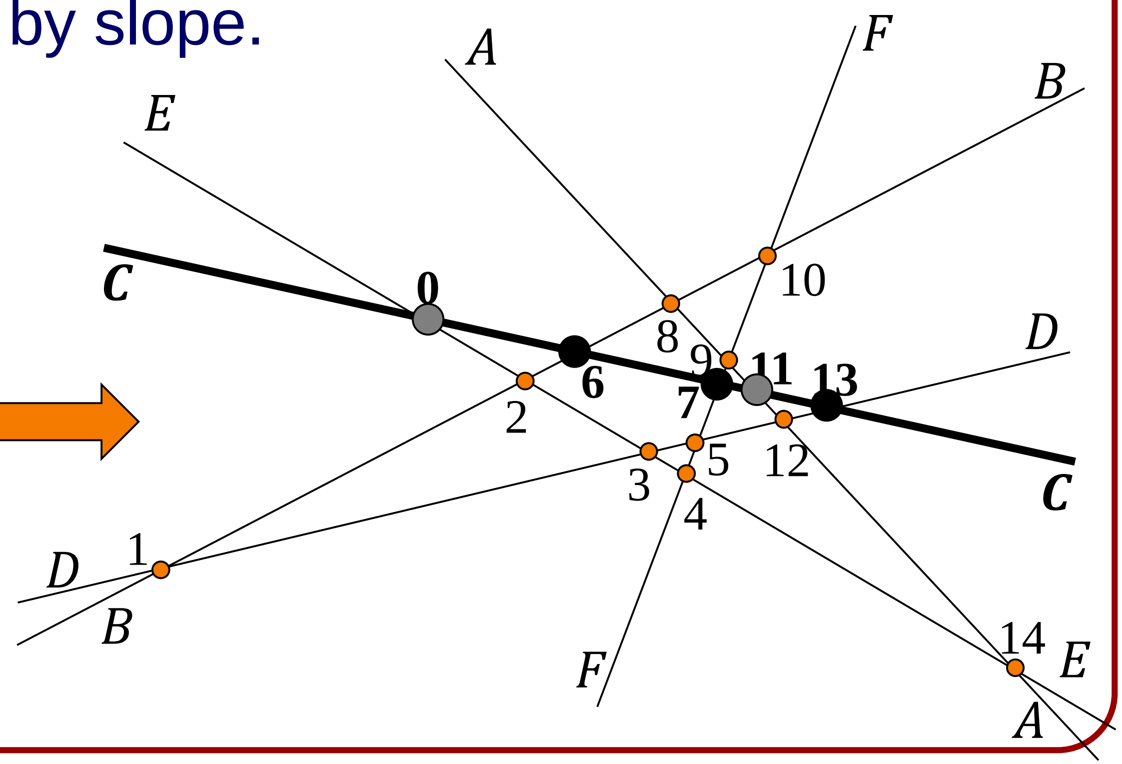


Visibility Graph [Welzl 1985]

For a line in the dual arrangement, the index of vertices on that line, left-to-right, increases.

⇒ Traversing the vertices the dual arrangement in sort-order, the slopes encountered around any primal vertex are sorted by slope.





Visibility Graph [Welzl 1985]

For any dual line, the sort order of vertices along that line, left-to-right, is increasing.

⇒ As we circle around vertex u, when we encounter segment (u, v), we will have already processed the edges out of v with smaller slope.