Mm5023 lecture 12

Optimization

Plan

- Weighted graphs
- Shortest path algorithm
- Minimal spanning trees (Kruskal and Prim)

Weighted graphs

A weighted graph is a pair (G, w) with G a graph or a multi graph and

$$w : E \longrightarrow \mathbb{R}_{>0}$$

a function weight function.

Example

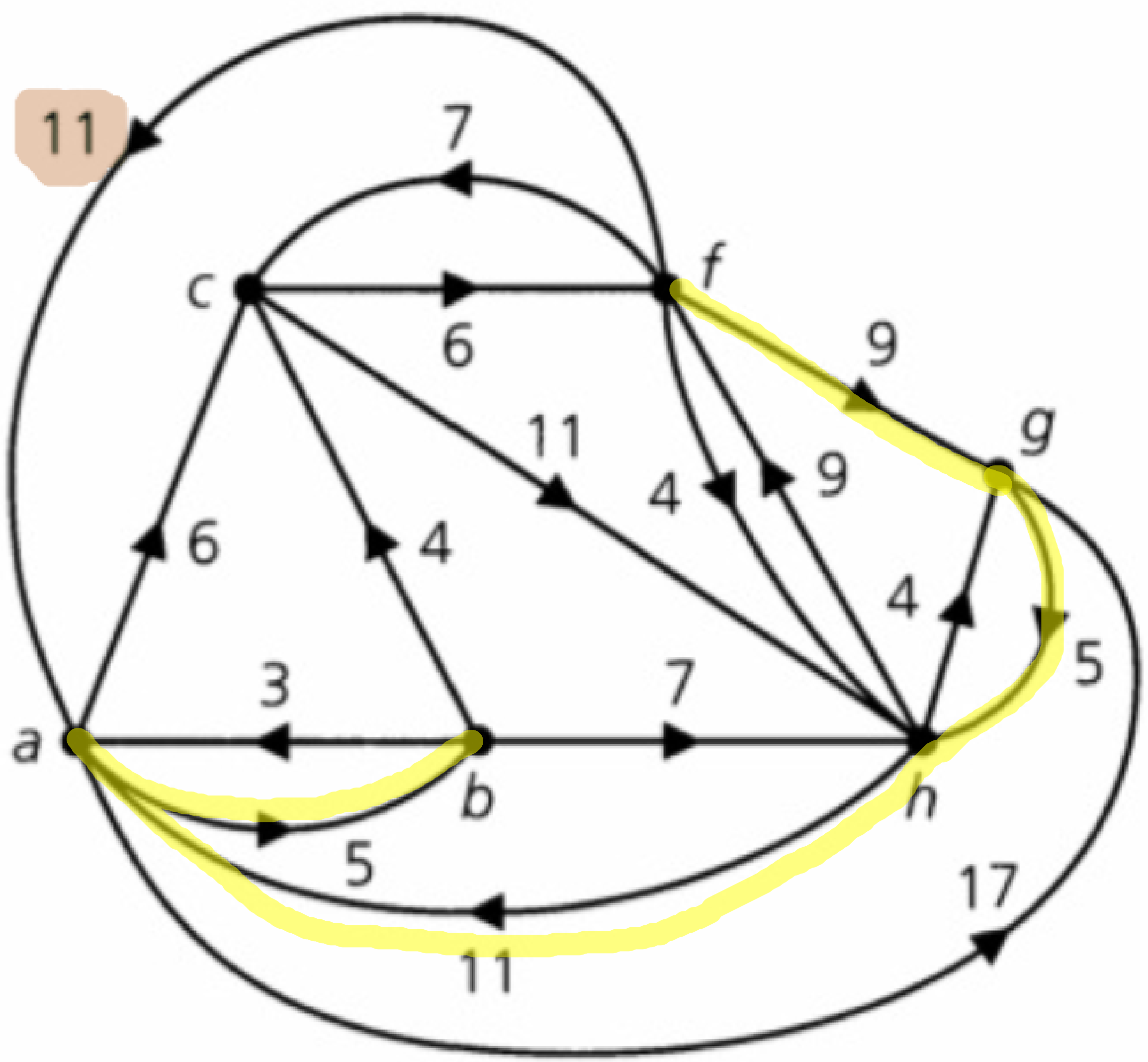


Figure 13.1

directed weighted graph

$w(f,a) = 11$

$w(f,c) = 7$

$w(c,f) = 6$

$l(p) = 9 + 5 + 11 + 5$
$= 30$

Length of a path

Given a weighted graph (G, w)
and a path $p := (v_0 \ldots v_n)$
the length of the path is

$$l(P) = \sum_{i=1}^{n} w(v_{i-1}, v_i)$$

sum of the weight of the edges that have been walked on

$p = (f\ a\ c\ h)$

$l(p) = w(f a) + w(a c) + w(c h)$

$= 11 + 6 + 11$

$= 28$

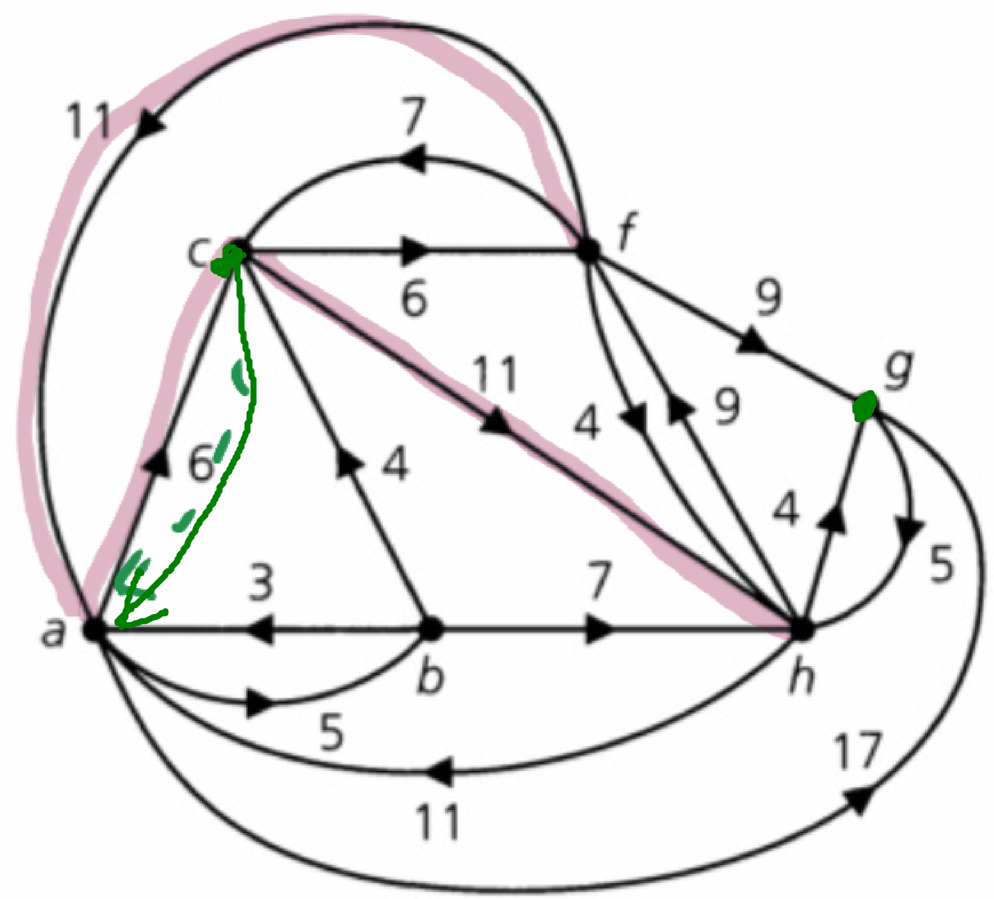


Figure 13.1

$w(c, a) = +\infty$

$w(c g) = +\infty$

$w(a c) = 6$

Convention: G directed

$$\omega : E(G) \longrightarrow \mathbb{R}_{>0}$$

You can extend ω to a function (that we call again ω)

$$\omega : V \times V \longrightarrow \mathbb{R}_{>0} \cup \{+\infty\}$$

$$\omega(v,u) = \begin{cases} \omega(v,u) & \text{if } (v,u) \in E(G) \quad (v \sim u) \\ +\infty & \text{if } (v,u) \notin E(G \end{cases}$$

G undirected you extend ω to the whole $\{ A \in \mathcal{P}(V) \mid |A| = 1, 2 \}$

$$\omega(A) = \begin{cases} \omega(A) & \text{if } A \in E(G) \\ +\infty & \text{otherwise} \end{cases}$$

G multi graph $\rightarrow$ similar

We can define $\mathbb{R}_{\geq 0} \cup \{+\infty\}$

$$d : V(G) \times V(G) \longrightarrow \overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$$

$$d(v,w) = \underline{\inf} \left\{ l(p) \;\middle|\; \begin{array}{c} p \text{ path from} \\ v \text{ to } w \end{array} \right\}$$

if v w are in the same conn component otherwise $\rightarrow$ if the graph is finite it is a $\boxed{\min}$

$d(v\ w) = +\infty$

$\inf \emptyset = +\infty$

If G is undirected then d defines a $\boxed{\text{pseudo}}$ - distance

1) $d(v,w) = 0$ iff $v = w$

2) $d(v,w) = d(w,v)$

$d(v,w)$ can be $+\infty$

3) $d(v\,w) \leq d(v\,t) + d(t,w)$

(Proof as Homework)

X set a distance is function

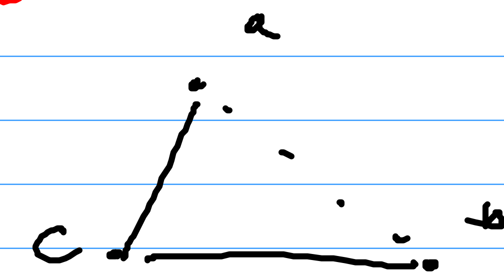
$$d : X \times X \longrightarrow \boxed{\mathbb{R}_{\geq 0}}$$

such that

d1) $, \quad d(x,y) = 0 \iff x = y$

not true in a directed graph

d2) $d(x,y) = d(y,x)$

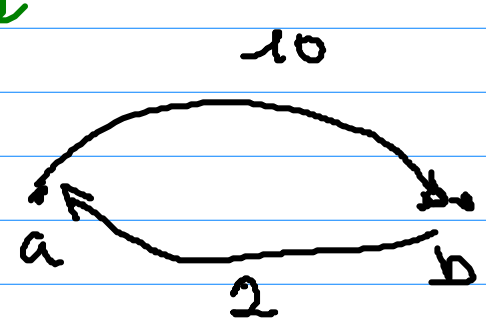


d3) (triangle inequality)

$$d(x,y) \leq d(x,z) + d(z,y)$$

The $d : V(G) \times V(G)$ just defined $d : V(G) \times V(G) \longrightarrow \mathbb{R}_{\geq 0} \cup \{+\infty\}$

Example



$d(a,b) = 10$

$d(b,a) = 2$

Rmk: if G is connected, undirected d is a distance

Sketch of the proof d is a pseudo distance G finite

1) $d(v,w)$ is $0 < \infty$ there is a path connecting v w

$$p = (v_1 = v, v_2 \ \dots \ v_n = w)$$

$$w(v_i v_{i+1}) > 0$$

$$l(p) = \sum_{i=1}^{n-1} w(v_i v_n)$$

if $n > 0$ $l(p) > 0$ $\min l(p) > 0$ unless $v = w$

2) path from $v \to w$ of minimal length $d(v,w)$ you reverse the path you get a path $w \to v$

$$d(w v) \le d(v,w)$$

Reverse the roles of v and w

$d(v,w) \le d(w,v)$

3) p_1 $(x \longrightarrow z)$

p_2 $(z \longrightarrow y)$

$p_1 p_2$ $(x \longrightarrow y)$

$d(x,y) \le l(p_1) + l(p_2)$ inf on right hau

$d(x,y) \le d(x,z) + d(z,y)$

Convention

We can extend $\omega : E \longrightarrow \mathbb{R}_{>0}$ to a function

$$\omega : V \times V \longrightarrow \overline{\mathbb{R}_{>0}} = \mathbb{R}_{>0} \cup \{+\infty\}$$

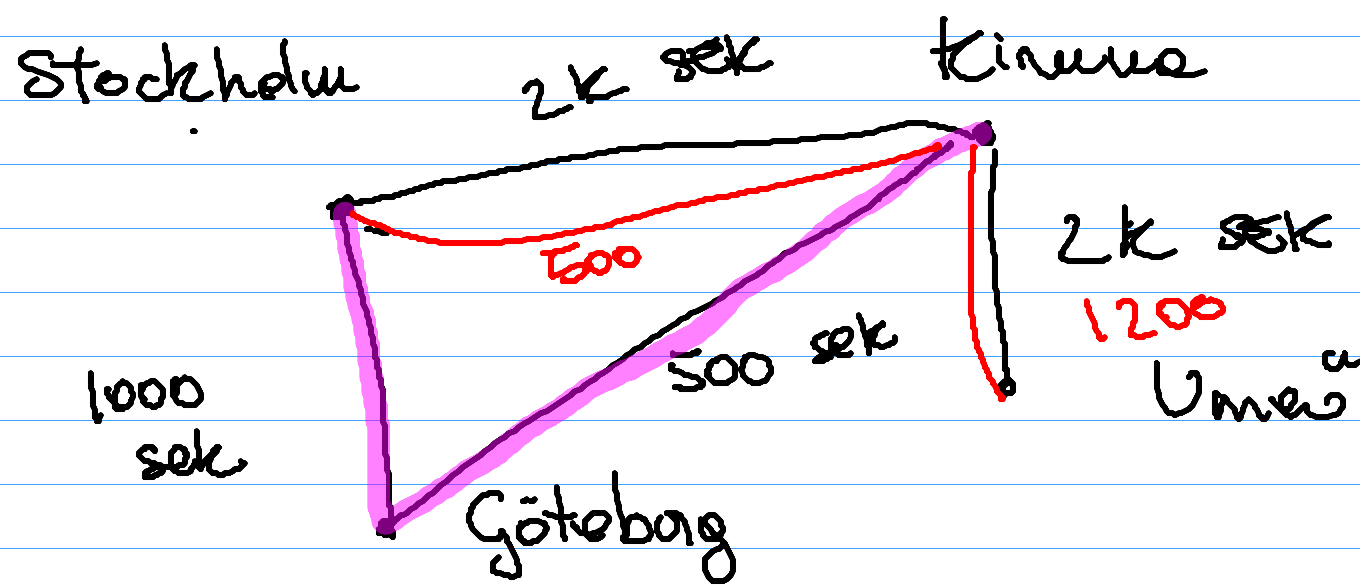
$$\omega(v,w) = \begin{cases} \omega\{vw\} & \text{if } v \sim w \\ +\infty & \text{otherwise.} \end{cases}$$

Problem G graph

$v, w \in V(G)$

Want the shortest path from v to w

Why this is usefull



want the cheapest travel from Stockholm to Umeå

Dijkstra Shortest path (finite graph)

INPUT: (G, w) weighted graph

$v \in V(G)$

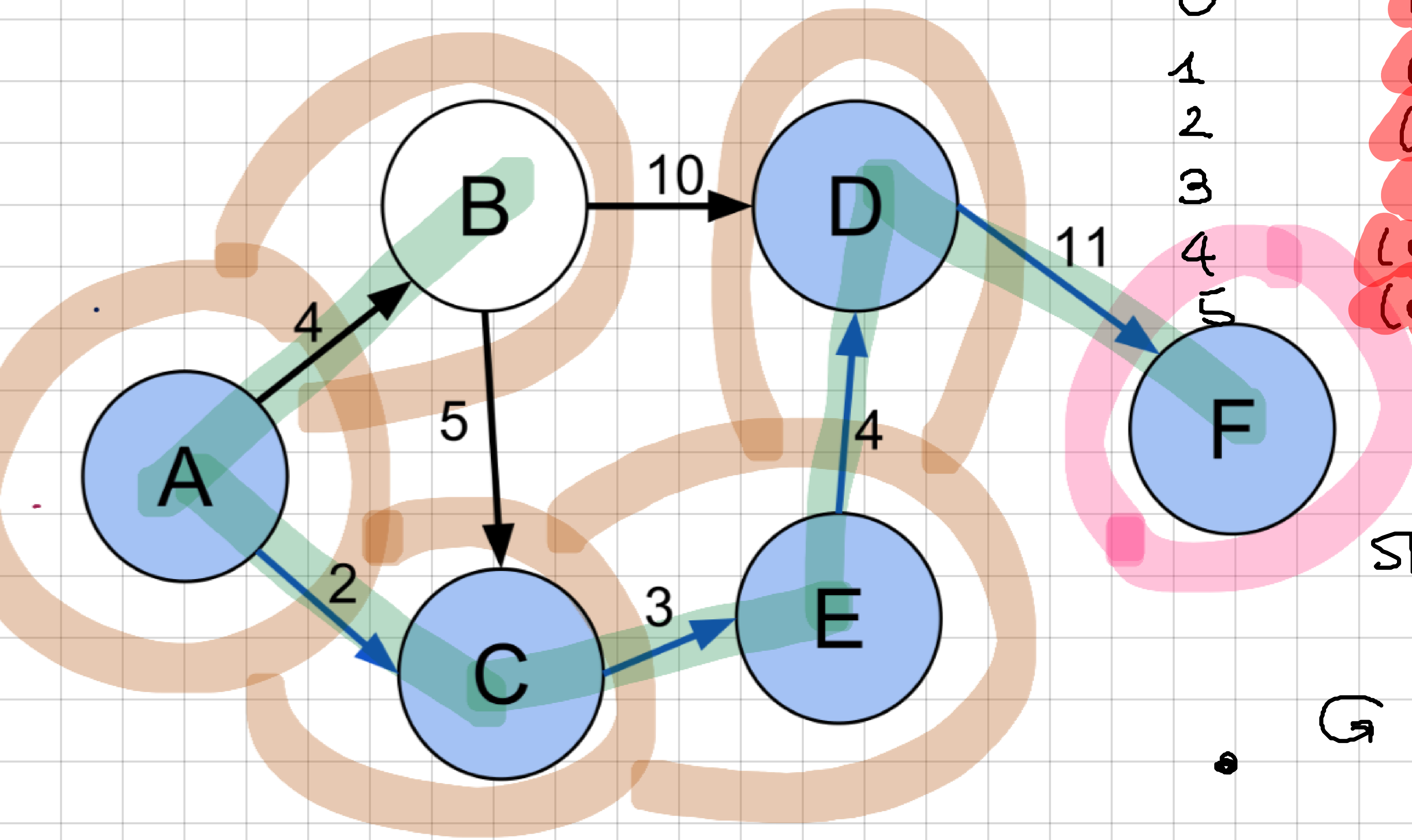
OUTPUT: a "labeled" weighted graph.

each vertex s has a label $(L(s), y)$

- $L(s) = d(v, s)$
- y preceding vertex in the shortest path.

Example

$v = A$



	A	B	C	D	E	F
0	$(0 -)$	$(+\infty -)$	$(+\infty -)$	$(+\infty -)$	$(+\infty -)$	$(+\infty, -)$
1	$(0 -)$	$(4, A)$	$(2 A)$	$(+\infty -)$	$(+\infty -)$	$(+\infty -)$
2	$(0 -)$	$(4, A)$	$(2 A)$	$(+\infty -)$	$(5 C)$	$(+\infty -)$
3	$(0 -)$	$(4, A)$	$(2, A)$	$(14, B)$	$(5 C)$	$(+\infty -)$
4	$(0, -)$	$(4, A)$	$(2, A)$	$(9 E$	$(5 C)$	$(+\infty -)$
5	$(0 -)$	$(4 A)$	$(2 A)$	$(9 E)$	$(5 C)$	$(20 D)$

$d(A, F) = L(F) = 20$

Shortest path

$(A\ C\ E\ D\ F)$

G

$n = |V(G)|$

Initialization

$i = 1$ $\quad S = \{v\}$ $\quad \bar{S} = V \setminus S = V \setminus \{v\}$

$L(v) = (0, -)$

$L(w) = (+\infty, -)$ for all $w \in \bar{S}$

If $n = 1$ Exit

Else go to step 2: For $i = 1 \ldots n-1$

For $w \in \bar{S}$

$$\lambda(w) \begin{pmatrix} L(w) \leftarrow \min_{u \in S} \{ L(w), L(u) + w(w,u) \} \\ y \in S \text{ is a vertex when } L(u) + w(w,u) \text{ reach a minim.} \end{pmatrix}$$

If for all $w \in \bar{S}$ $\lambda(w) = \begin{pmatrix} +\infty \\ * \end{pmatrix}$ Exit

(we have processed the connected cp of v & cannot go further.)

ELSE let $w \in \bar{S}$ such that $L(w)$ is minima $\quad S \leftarrow S \cup \{w\}$

$i \leftarrow i+1$ $\quad \bar{S} \leftarrow \bar{S} \setminus \{w\}$

As mathematicians algorithm requires proofs of the fact that they work.

→ TERMINATION

- Exit condition if G is not connected
- S grows at every step if G is connected

→ $\boxed{\text{OUTPUT IS CORRECT}}$ proved by induction.

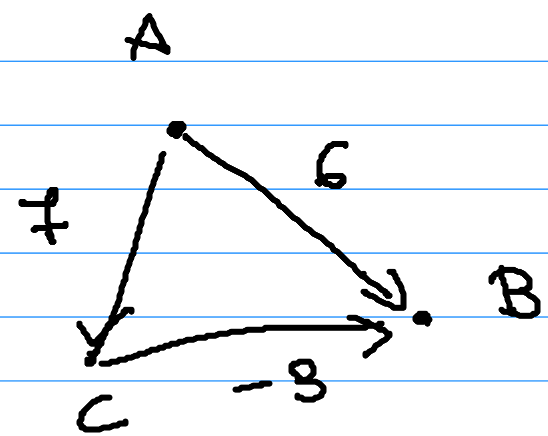
Minimal spanning tree

Let $(G=(V,E), w)$ a weigthed undirected graph a minimal spanning tree is a spanning tree T such that

$$\underbrace{w(T)}_{\text{weigth of } T} := \sum_{e \in E(T)} w(e)$$

is minimal

The algorithm does not work if you have negative weight !



	A	B	C
..	(0 -)	(+∞ -)	(+∞ -)
	(0 -)	(6 A)	(7 C)
	(0 -)	(6 A)	(7 C)
		~~(4, C)~~	

Minimal Spanning trees

Def (G, w) weighted undirected graph. a minimal spanning tree is a spanning tree $T \subseteq G$

st

$$w(T) := \sum_{e \in E(T)} w(e)$$

is minimal.

Kruskal

at every iteration you might get a forest

Prim

you might get a tree

* allow for graph be encoded by matrices

$$\begin{array}{c|ccc} & v_1 & \cdots & v_n \\ \hline v_1 & 0 & \cdot & \\ \vdots & & & \\ v_n & & & \end{array}$$

a_{ij} $w(v_i v_j)$

GREEDY you minimize at every step

characterizations of trees

Kruskal's Algorithm

Step 1: Set the counter $i = 1$ and select an edge e_1 in G, where $\text{wt}(e_1)$ is as small as possible. $E(T) = \{e_1\}$

Step 2: For $1 \leq i \leq n - 2$, if edges $e_1, e_2, \ldots, e_i$ have been selected, then select edge e_{i+1} from the remaining edges in G so that (a) $\text{wt}(e_{i+1})$ is as small as possible and (b) the subgraph of G determined by the edges $e_1, e_2, \ldots, e_i, e_{i+1}$ (and the vertices they are incident with) contains no cycles.

Step 3: Replace i by $i + 1$.

If $\boxed{i = n - 1}$, the subgraph of G determined by edges $e_1, e_2, \ldots, e_{n-1}$ is connected with n vertices and $n - 1$ edges, and is an optimal spanning tree for G.

If $i < n - 1$, return to step (2).

$n = |V(G)|$

T spanning tree then $|E(T)| = n - 1$

Example

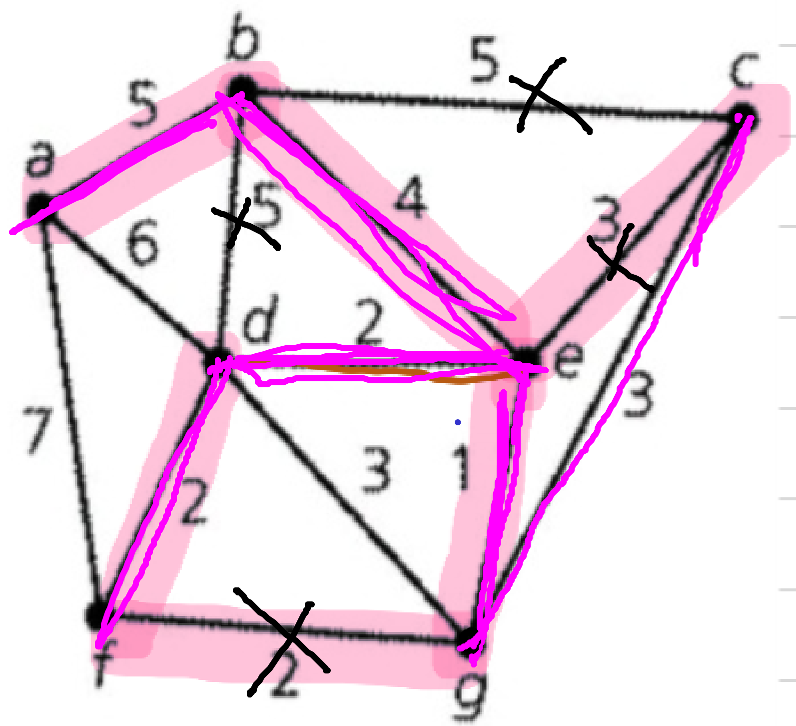


Figure 13.5

0 $E(T) = \{\{eg\}\}$

1 $E(T) = \{\{eg\}\ \{ef\}\}$

$E(T) = \{\{e,g\}\ \{ef\}, \{ed\}\}$

$E(T) = \{ \text{———} \{gc\}\}$

$E(T) = E(T) \cup \{\{be\}\}$

$E(T) = E(T) \cup \{ab\}$

$E(T)$

$w(T) = \underline{1+2+2} + 3 + 4 + \underline{5} = 17$

$P \quad \overline{P} \overset{=N}{\subseteq} V(G) \quad T$

Prim's Algorithm

Step 1: Set the counter $i = 1$ and place an arbitrary vertex $\boxed{v_1 \in V}$ into set P. Define $N = V - \{v_1\}$ and $T = \emptyset$.

Step 2: For $1 \leq i \leq n - 1$, where $|V| = n$, let $P = \{v_1, v_2, \ldots, v_i\}$, $T = \{e_1, e_2, \ldots, e_{i-1}\}$, and $N = V - P$. Add to T a shortest edge (an edge of minimal weight) in G that connects a vertex x in P with a vertex y $(= v_{i+1})$ in N. Place y in P and delete it from N.

(P, T) is a tree at any iteration.

Step 3: Increase the counter by 1.

If $i = n$, the subgraph of G determined by the edges $e_1, e_2, \ldots, e_{n-1}$ is connected with n vertices and $n - 1$ edges and is an optimal tree for G. Tree by the characterization

If $i < n$, return to step (2).

$i \leftarrow i+1$

Spanning $V(T) = V$

Example

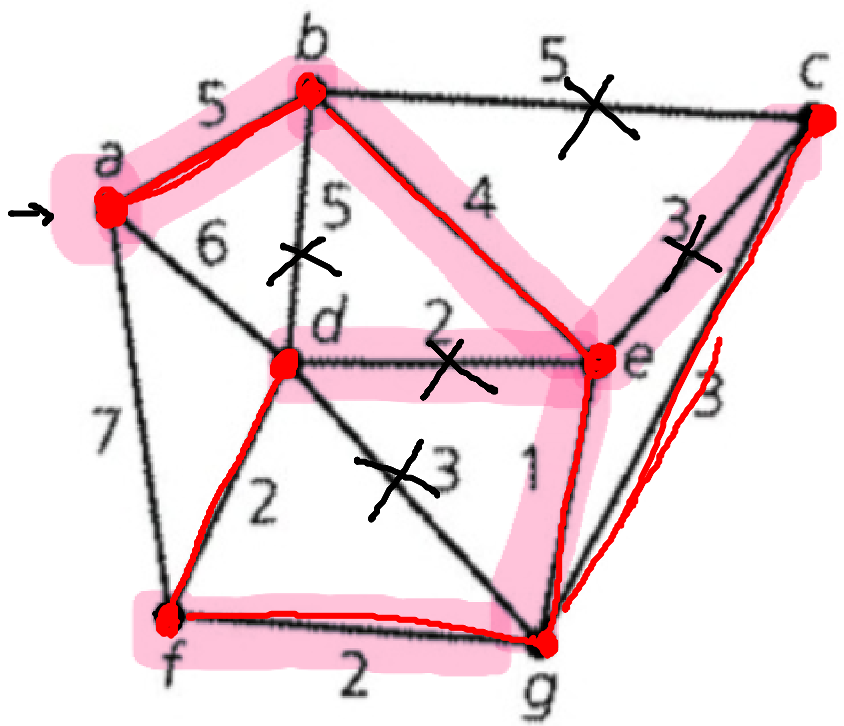


Figure 13.5

initialization $P = \{f\}$ $N - \overline{P}$

$E(T) = \emptyset$

1) $P = \{f, d\}$ $T = \{\{f, d\}\}$ $N = \overline{P}$

2) $P = \{f, d, g\}$ $T = T \cup \{\{fg\}\}$

3) $P = \{f, d, e\}$ $T = T \cup \{\{ge\}\}$

4) $P = \{f, d, e, c\}$ $T = T \cup \{\{cg\}\}$

5) $P = \{f d e c b\}$ $T = T \cup \{\{eb\}\}$

6) $P = \{f d e c b a\}$ $T = T \cup \{\{ab\}\}$

$w(T) = 2 + 2 + 1 + 3 + 0 + 5 = 17$

As before !!

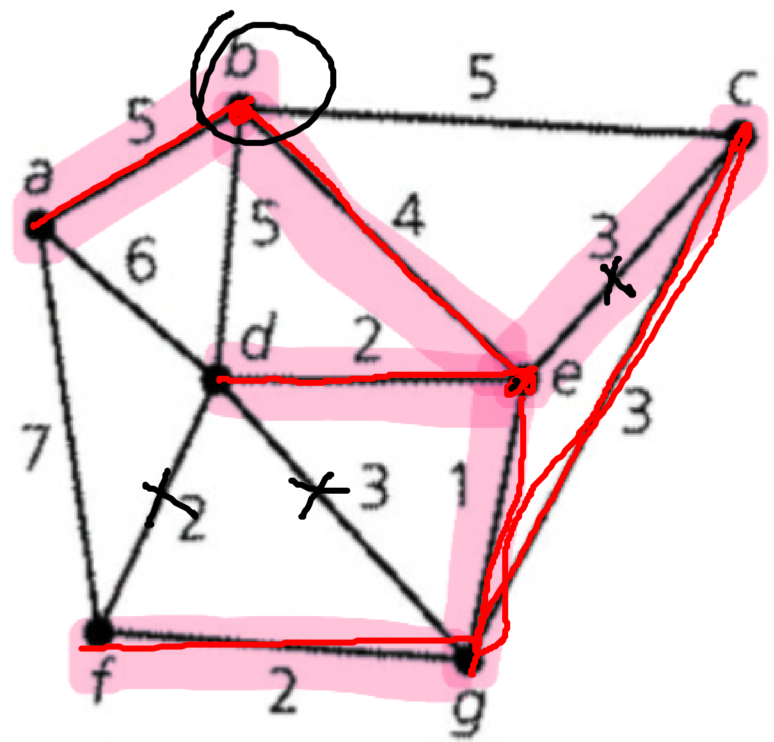


Figure 13.5

Starting vertex e

$E(T) = \emptyset$ $\quad$ $P = \{e\}$ $\quad$ $w(T) = 0$

1) $P = \{e, g\}$ $\quad$ $E(T) = \{\{eg\}\}$ $\quad$ $w(T) = 1$

2) $P = \{e\, g\, d\}$ $\quad$ $E(T) = E(T) \cup \{\{ed\}\}$ $\quad$ $w(T) = 3$

3) $P = \{e\, g\, d\, f\}$ $\quad$ $E(T) = E(T) \cup \{fg\}$ $\quad$ $w(T) = 5$

4) $P = \{e\, g\, d\, f\, c\}$ $\quad$ $E(T) = E(T) \cup \{\{cg\}\}$ $\quad$ $w(T) = 8$

5) $P = \{e\, g\, d\, f\, c\, b\}$ ——— $\quad$ $w(T) = 12$

6) $P = \{e\, g\, d\, f\, c\, b\, a\}$ $\quad$ $E(T)$ ——— $\quad$ $w(T) = 17$

Important : in the exam if you are asked to compute a minimal spanning tree with a given algorithm you need to show all the iterations

The same is for shortest path : Show the iteration table

Let us see that Kruskal algorithm does what is suppose to.

it stops   the output is a graph with no cycle and n vertices and $n-1$ edges
$\longrightarrow$ tree.
Forest which is not a tree would have less than $n-1$ edges

$E(T) = \{e_1 \quad e_n\}$ where the edges are enumerated following the order in which they have been chosen

T' a minimal spanning tree (it exist by the well ordering G finite)
such that
$d(T') = \max \{ i \in \{1 \ldots n\} \mid \{e_1 \ldots e_i\} \in T' \}$ is maximum

we want $T = T'$ & ($\Leftrightarrow d = n-1$) T is a minimal spanning tree

Assume by contradiction $d < n-1$ then $T' + e_{d+1}$ contains a cycle $e_{d+1} \in E(T')$

$\Rightarrow \exists\ e \in E(T')$ wich is not $E(T)$

$T'' = T' + e_{d+1} - e$ spanning tree

$\{e_1 \dots e_d, e\} \subseteq E(T')$

$\hookrightarrow$ give me a forest

$\omega(e) \geq \omega(e_{d+1})$ by the choice of the algorith.

$$\underset{T'_{min}}{\omega(T')} \leq \omega(T'') = \omega(T') + \underbrace{\omega(e_{d+1}) - \omega(e)}_{\leq 0} \leq \omega(T')$$

$\omega(T'') = \omega(T')$ T'' minimal spanning tree

$d(T'') \gneq d(T')$ contradicting the choice of T'

$\Rightarrow T' = T$