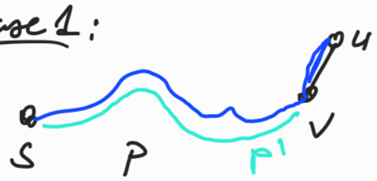
5 Graphs

dist_BFS

L.1. Let $G=(V,E)$ be a graph & $s \in V$.
Then $d(s,v) \leq d(s,u)+1 \quad \forall \{u,v\} \in E$.

proof: if v no sv-path $\Rightarrow$ no vu path $\Rightarrow d(sv) = d(su) = \infty$.

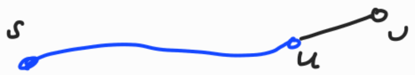
else **case 1:**



shortest su path contains $v \Rightarrow d(su) = d(sv)+1$

as p' must be shortest sv path

Case 2 shortest su path does not contain v



$\Rightarrow d(sv) \leq d(su)+1 \quad \square$

L.2: Let $G = (V, E)$ be a graph.
Then, for all $u \in V$ the value $u.d \geq d(s,u)$
at all steps of dist_BFS(G, s)
[incl. termination].

proof:

proof by induction on # enqueue-operations.

Base case: after $\underline{1}$ enque-op (L.4): $s.d = 0 = d(s,s)$
& $u.d = \infty \geq d(s,u)$ ✓

Assume statement true after k enqueue-operations.
Now, the k+1 enqueue-op takes place
in FOR-loop for some $w \in N(v)$: Q.enqueue(w)

Before Q.enqueue(w): $u.d \geq d(s,u) \forall u$ by ind. hyp.
Then in L8: $w.d = v.d + 1 \geq d(s,v) + 1 \geq d(s,w)$

by ind. hyp.

since $\{v,w\}$ edge & by L.1

Since all other u.d remain unaffected & $w.d \geq d(s,w)$
$\Rightarrow$ AFTER Q.enque(w) (= k+1 enqueue op)
$\forall u \in V$: $u.d \geq d(s,u)$

Since each u is when enqued once
never enqued again (since marked as visited)
$\Rightarrow$ value u.d remains unchaged after
u enqueued once $\Rightarrow$ $\square$

L.3 Suppose during run of dist_BFS on $G=(V,E)$, the queue $Q=(v_1,\dots,v_r)$
first $\uparrow v_1$, last $\uparrow v_r$

Then, i) $v_r.d \le v_1.d+1$ & ii) $v_i.d \le v_{i+1}.d$, $1\le i<r$

[intuition: $\overbrace{v_1.d \le v_2.d \le \dots \le v_r.d}^{(ii)} \le v_1.d+1$ (i)
dist-values of all vertices are either all equal or form sequence $(k,..k,k+1..k+1)$ for some integer $k\ge 0$

proof: Now, induction over # operation on Q. (enqueue + dequeue)

Base case: # op on $Q=1$ $\Rightarrow$ $Q=(s)$ in L4 statement trivially true as Q contains only 1 element.

Assumption: statement true after #op on $Q=k$.

$k+1$ op. can be dequeue or enqueue.

Suppose first $k+1$ op on $Q=(v_1\dots v_r)$ is dequeue

Ind hyp: (i) $v_r.d \le v_1.d+1$ & $*$
(ii) $v_1.d \le v_2.d$ $**$

$\rightarrow$ after dequeue: $Q=(v_2\dots v_r)$ or $Q=\emptyset$ if $r=1$
[must show i) ii) holds for this Q'] i, ii trivially true

by $*$ $v_r \le v_1.d+1 \overset{**}{\le} v_2.d+1$ $\Rightarrow$ (i) $v_r \le v_2.d+1$
& (ii) holds for all $2\le i\le r$

$\Rightarrow$ i) ii) satisfied for Q.

Now assume $k+1$ op. on $Q = (v_1 \dots v_r)$ is enqueue (L. 9)

$\underline{k+1 \text{ op}:}$ $Q.\text{enqueue}(w) \Rightarrow \tilde{Q} = (v_1 \dots v_r\ w)$

[must show i) ii) holds for this $\tilde{Q}$]

if $Q = (\ )$ before enqueue w, i & ii) trivially hold.
as $\tilde{Q} = (w)$

Ass $r \geq 1$.

Before FOR-loop where w is is enqueue
vertex v was dequeued [$v \in N(w)$] in L.6

In particular, since v was already in the queue
& in L6 dequeued directly before FOR-loop where $w \in N(v)$
is taken.

<u>Before v dequeued</u>

$$\tilde{Q} = (v, x_1 \dots x_{r'})$$

$\hookrightarrow$ Ind. Hyp: $\underbrace{v.d \leq x_1.d \leq \dots \leq x_{r'}.d}_{(ii)} \leq v.d + 1$ (*)

(i) *

<u>after v dequeued:</u>

$$\tilde{\tilde{Q}} = (x_1 \dots x_{r'})$$

<u>after W endqueued.</u>

$$Q = (\overset{v_1}{x_1} \dots \overset{v_{r'}}{x_{r'}}, \overset{v_{r'+1}}{y} \dots \overset{v_{r-1}}{} \overset{v_r}{W})$$

by construction (L8)

$w.d = v.d + 1$

$\overset{(*)}{\leq} x_1.d + 1$

$\Rightarrow$ <u>i) holds for Q.</u>

For ii) observe:

$$Q = \left(\overset{v_1}{x_1} \ldots \overset{v_{r'}}{x_{r'}}, \overbrace{\overset{v_{r'+1}}{y} \ldots \overset{v_{r-1}}{} \overset{v_r}{w}}^{\text{all those must be in } N(v)!} \right)$$

Line 8 $\Rightarrow v_{r'+1}.d = \ldots = v_{r-1}.d = \overset{w.d}{\overset{\shortparallel}{v_r.d}} = v.d+1$

(*)

$$v.d \leq \underset{v_1 d}{x_1.d} \leq \ldots \leq \underset{v_{r'}.d}{x_{r'}.d} \leq v.d+1 = v_{r'+1}.d = \ldots = v_{r-1}.d = \underset{\underset{w.d}{\shortparallel}}{v_r.d}$$

$\Rightarrow$ ii) holds for Q.

✓ $\square$

COR 1: IF v is enqued before v' is enqued in distBFS.
THEN $v.d \leq v'.d$ (direct consequence of L3.)

Thm: dist_BFS(G,s) computes all dist $d(s,v)$ correctly for all $v \in V(G)$: $v.d = d(s,v)$ after termination.

proof:

By contradiction, assume $v.d \neq d(s,v)$.

Choose v among all vertices v' with $v'.d \neq d(s,v')$ as the one for which $d(sv)$ is minimum. $\Big\}\circledast$

Lemma 2 + $v.d \neq d(sv)$ implies: $v.d > d(s,v)$

Note $v \neq s$, since $s.d = d(s,s) = 0$ correct (L3)

Moreover there must be an sv-path in G, otherwise $\infty = d(s.v) \geq v.d$ (contradicting $v.d > d(s,v)$)

Let P be some shortest sv-path $(s \dots \underbrace{uv}_{\text{edge}})$
(since $s \neq v$, u exists where $u=s$ possible)

$\Rightarrow d(sv) = d(su) + 1$

By choice of v (min dist $\circledast$) $\Rightarrow \underbrace{u.d = d(s,u)}_{<\infty}$

$\Rightarrow v.d > d(sv) = d(su) + 1 = u.d + 1 \quad \circledast$

Now consider time when distBFS choses

"$u := Q.\text{dequeue}$"

which must happen as $u.d < \infty$

v is either visited or not at this point.

<u>IF v is non visited</u> $\Rightarrow$ $v \in N(u)$ & FORLOOP (L7)
implies v will be considered
& $v.d = u.d + 1$ ↯
contradiction to (*)

<u>IF v visited</u> $\Rightarrow$ then it was enqued
to Q at some point (L9+10)

▶ IF v was enqued before u was enqued

$\Rightarrow$ COR 1: $v.d \leq u.d$ ↯ contradiction to (*)

▶ IF v was enqued after u was enqued

$\Rightarrow$ $v \in N(u')$, $u' \neq u$, u' was enqued before u.

& $v.d = u'.d + 1$

$\Rightarrow$ COR 1: $u'.d \leq u.d$ $\Rightarrow$ $v.d \leq u.d + 1$ ↯
contr. (*)

$\Rightarrow \quad v.d = d(s,v) \quad \forall v \in V$

[this also implies additionally that all vertices reachable from s are visited otherwise $v.d = \infty > d(s,v)$ would hold]

$\square$

KRUSKAL

Theorem: Kruskal's algorithm correctly computes a MST for given undirected graph G. in $O(|V||E|)$ time

- **RUNTIME [sketch]:**

Sort edges : $O(|E| \log(|E|)) = O(|E| \log |V|)$ since $|E| \leq |V|^2$ $\Rightarrow \log(|E|) \leq \log(|V|^2) = 2 \log(|V|)$

$F = \emptyset$: $O(1)$

$T = (V, F)$: $O(|V|)$

FOR loop: in the i-th step, when you check if $(V, F \cup \{uv\}) = T'$ is acyclic

Start $BFS(T', v)$ $\Rightarrow$ this goes along all vertices that are reachable from v

$\Rightarrow$ gives you incl.-max subgraph H of G that contains v & any new cycle is in A (if there is any)

H as at most $2i$ vertices & i edges.

$\Rightarrow BFS(T', v)$ in i-th step runs in $O(i + 2i) = O(i) \leq O(|V|)$ time

Total: $O(|E||V|)$

with efficient datastructure, this can be improved to $O(|E| \log |V|)$

- **Correctness?**

proof:

1) Alg. terminates ✓

2) resulting graph T contains is acyclic & $V(T) = V(G)$ } T is spanning forest of G.

3) T is connected: IF not $\Rightarrow \exists\, x,y \in V(T)$ st no x-y-path exists in T

G is connected $\Rightarrow \exists$ x-y-path in G.

$\Rightarrow$ simple x-y-path & edge one st $e \notin F$:

in G

at some ith step of FOR-loop edge e is considered.

But $T' = (V, F \cup \{e\})$ remains acyclic (otherwise there would have been a x-y-path already in T' & thus in T)
(at step i)

$\Rightarrow$ e is added to T at step i $\lightning$ $e \notin F$

$\Rightarrow$ T is spanning tree of G.

remains to show that T is <u>minimum</u> spanning tree of G.

Let F_i be the set of edges formed up to step i

<u>Claim:</u> $\exists$ MST T^* of G st $F_k \subseteq E(T^*)$. $\forall$ steps $k = 1..m$

$i = 0$, $T = (V, \emptyset)$ ✓

<u>Ind hyp:</u> Assume claim is true for all steps $i \leq k$

Let T^* be MST of G st $F_k \subseteq E(T^*)$

now: $i = k+1$. and let $e = (uv)$ be the current edge considered in step $i = k+1$, i.e.,

- IF e not added to F_k
  THEN $F_{k+1} = F_k \subseteq E(T^*)$ ✓

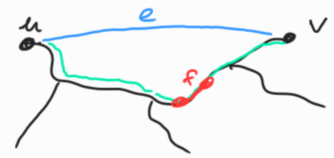
- IF e added to F_k
  THEN $F_{k+1} = F_k \cup \{e\}$

↳ IF $e \in E(T^*)$ $\Rightarrow$ $F_{k+1} \subseteq E(T^*)$ ✓

ELSE $e \notin E(T^*)$ & therefore, $(V, E^*(T) \cup \{e\})$ must contain a simple cycle

since $\forall\, uv\ \exists!$ simple uv-path in T^*, since $e = \{uv\}$ not in T^* $\Rightarrow$ $x \{\}_y + e$ forms cycle

Situation: T^*



unique simple u-v-path P in T^*

$e = (uv)$

Since e is contained in T, at least one edge f of P cannot be contained in T

otherwise T contains cycle.

$\Rightarrow \exists f \notin F_k \subseteq E(T)$

Now, $\tilde{T} =$ "$T^* - f + e$" is again a tree. (exercise)

Moreover, $w(f) \geq w(e)$ [otherwise if $w(f) < w(e)$, then the greedy-choice would be f & not e in step $i = k+1$]

$\Rightarrow w(\tilde{T}) \leq w(T^*)$

& since T^* is MST $\Rightarrow w(\tilde{T}) \geq w(T^*)$ $\Rightarrow$ $w(\tilde{T}) = w(T^*)$

$\Rightarrow$ $\tilde{T}$ is MST that contains the edges in F_{k+1}.

Now repeat the latter arguments until all edges of T are contained in MST $\Rightarrow$ T is MST $\square$