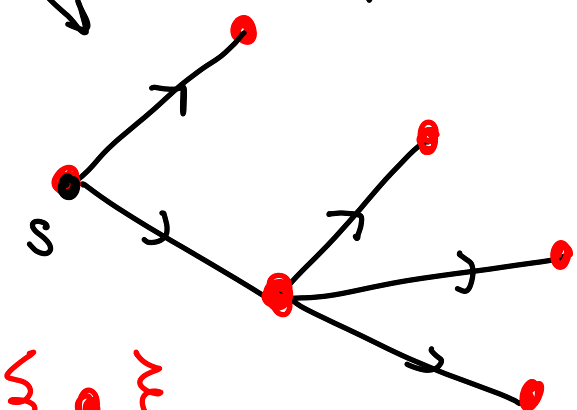
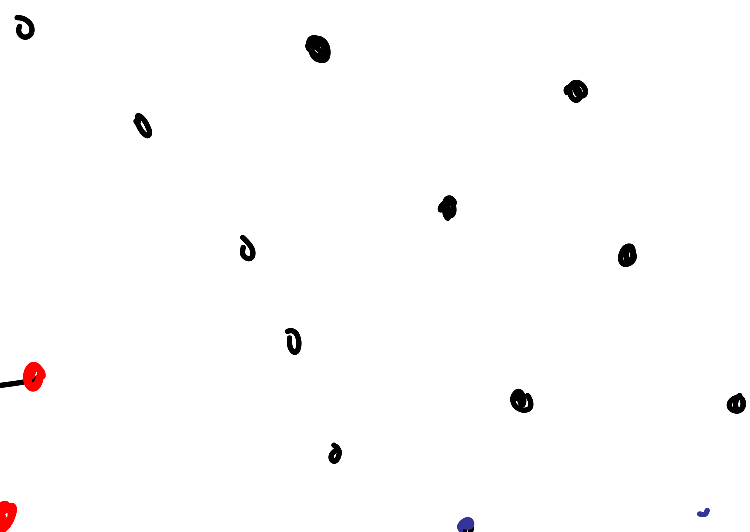
10/15/2007

Dijkstra's Algorithm

Now we assume that $\ell(e) \geq 0$ for $e \in E$.

$d(v)$ = length path s to v in tree





$S = \{ \bullet \}$

$S = \{s\}$ at start.

$v \in S$: $d(v)$ = shortest distance s to v.

P1

P2

$d(v)$ = min. length path ....>....v

one non-tree edge

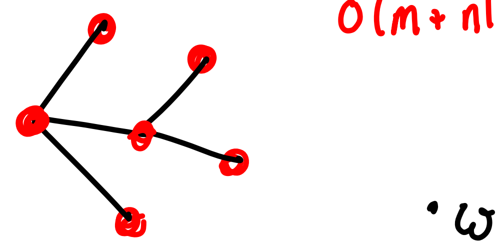
Running Time of Dijkstra: $O(n^2)$.

$O(m \log n)$ HEAP

$O(m + n \lg n)$

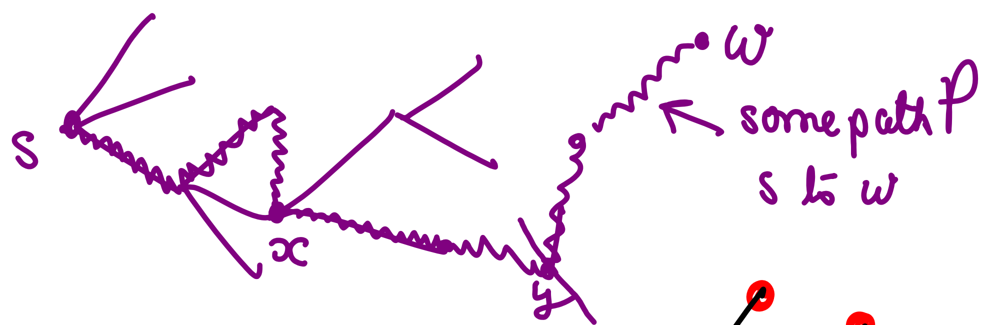
$$d(w) = \min_{v \notin S} d(v)$$

$$S \leftarrow S \cup \{w\}$$



P1

Claims that $d(w)$ is correct.

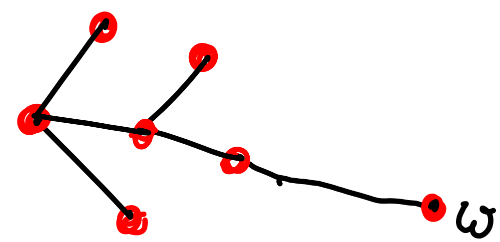


$\ell(P) \geq d(w)$

$\ell(P) \geq \ell[P:y]$

$\geq d(y)$

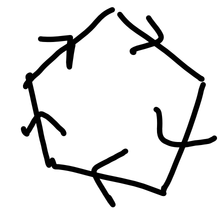
$\geq d(w)$

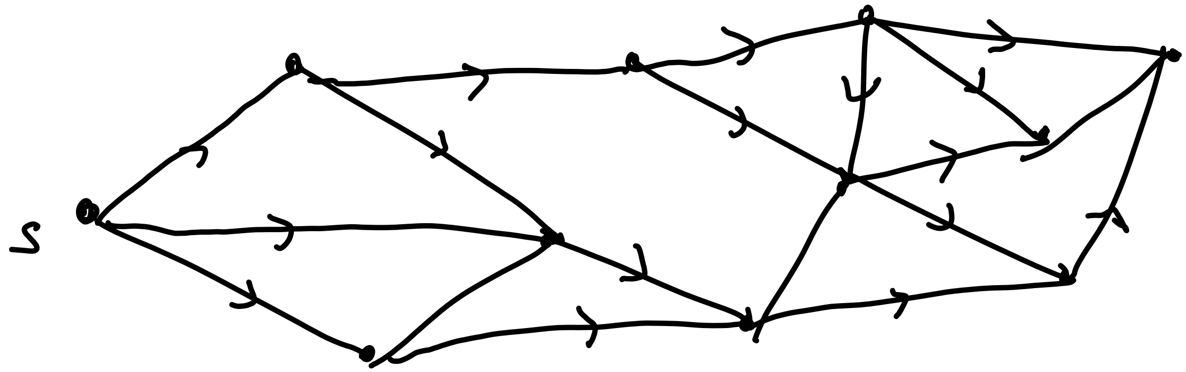


For $v \notin S \cup \{w\}$

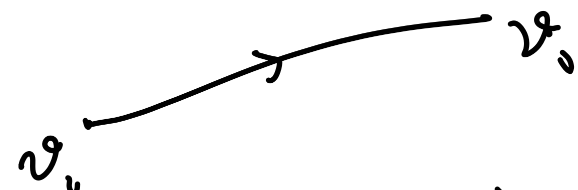
$$d(v) \leftarrow \min\{d(v), d(w) + \ell(wv)\}$$

preserves P2

DAG: Directed Acyclic Graph. No 



$O(m)$ algorithm.

Order vertices 

$i<j$

$$d(v_j) = \min_{i<j} d(i) + l(v_i, v_j)$$

All pairs shortest paths

Given l_{ij} = length of edge (i,j)

Find shortest path between every pair of vertices.

Floyds Algorithm

Initialise $d(i,j) = l_{ij}$

for $k = 1, 2, \dots n$

round k

for $i = 1, 2, \dots, n$

for $j = 1, 2, \dots, n$

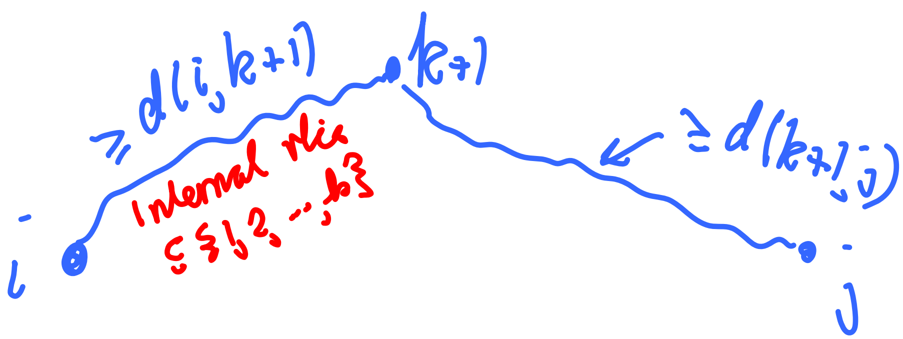
$$d(i,j) = \min\{ d(i,j), d(i,k) + d(k,j) \}$$

Claim:

At end of round k,

$d(i,j)$ = minimum length of a path from i to j whose internal vertices are in $\{1,2,\dots,k\}$

Proof: By induction on k. True for $k=0$.



Path using $\subseteq \{1,2,\dots,k+1\}$ — If $k+1$ not used $d(i,j) \leq \ell(P)$