Shortest Paths

Digraph with edge weights (costs, distances)

Shortest path from s to t: path of minimum total wt.

Problems:

single pair: given s, t, find a shortest path from s to t

single source: given s, find shortest paths from s to all reachable vertices

all pairs: find shortest paths between all pairs

Cases:

acyclic

no negative wts

general

(planar, etc.)

Properties:

$\exists$ a shortest path from s to t iff there is no negative (total wt.) cycle on a path from s to t. If there is no such cycle, there is a shortest path that is simple (no repeated vertex).

If no neg cycle reachable from s, then $\exists$ shortest path tree rooted at s, contains all vertices reachable from s, all tree paths are shortest paths in graph.

New goal: find a negative cycle or construct a shortest path tree.

(single-source problem is central)

Given a spanning tree T rooted at s,

$d(v)$ = tree wt from s to v, is T a shortest path tree?

Yes, iff there is no (v,w) with $d(v) + c(v,w) < d(w)$

Edge relaxation algorithm to find a shortest path tree:

$d(s) = 0$, $d(v) = \infty$ for $v \neq s$

while $\exists$ edge (v,w) with $d(v) + c(v,w) < d(w)$
do { $d(w) = d(v) + c(v,w)$; $p(w) = v$ }

$d(v)$ is always the wt of some s-v path

if algorithm stops and p defines a tree, must be a shortest path tree

stops iff no neg cycle

(alg maintains $d(w) \geq d(v) + c(v,w)$ if $v = p(w)$)

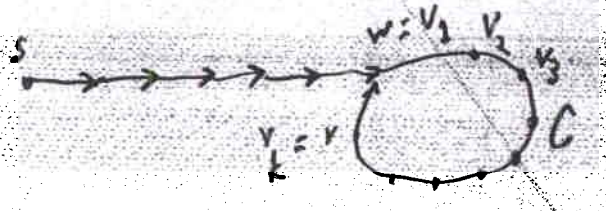
Suppose T not a sp tree. Let x be such that $d(x) >$ s-x distance. Let P be a shortest path from s to x, $d'(v)$ = P-distance from s, (v,w) first edge along P such that $d'(w) < d(w)$.

Then $d(v) + c(v,w) = d'(v) + c(v,w) = d'(w) < d(w)$.

(This gives the hard direction of sp tree test.)

Suppose edge relaxation algorithm creates a cycle.

Then it must be a negative cycle.



$d(v) + c(v,w) < d(w) \Rightarrow d(v) - d(w) + c(v,w) < 0$

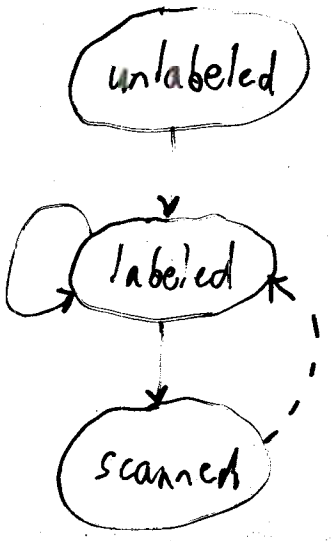
Sum around cycle: $\sum_{i=1}^{k} d(v_i) - d(v_{i+1}) + c(v_i, v_{i+1}) < 0$

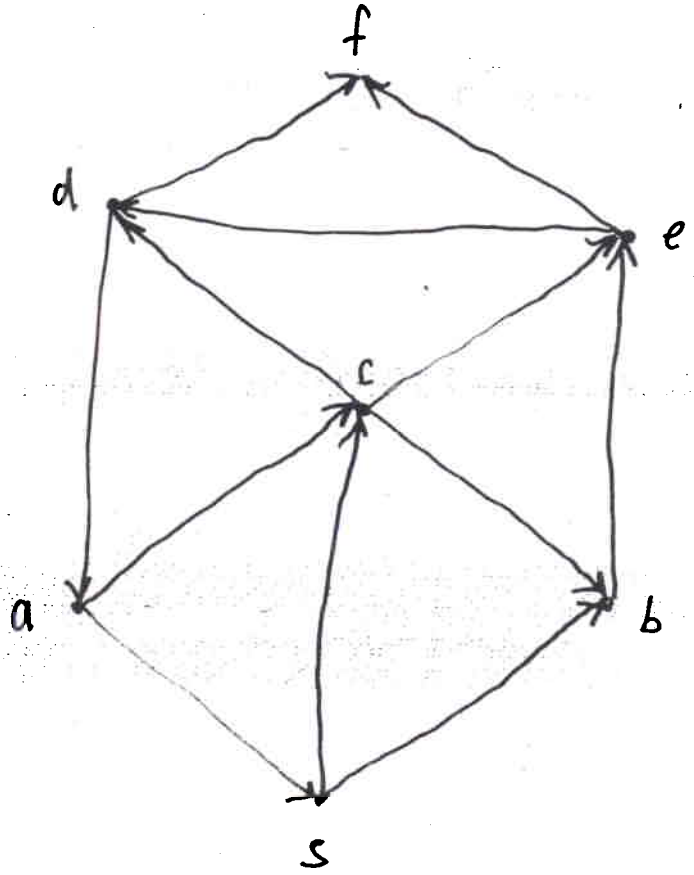
$\sum_{i=1}^{k} c(v_i, v_{i+1})$

Labeling and scanning algorithm:

$L = \{s\}$; $d(s) = 0$; $d(v) = \infty$ for $v \neq s$;

while $L \neq \emptyset$ do {
  remove v from L;
scan(v): for each (v, w) do
    if $d(v) + c(v, w) < d(w)$ then
      {$d(w) = d(v) + c(v, w)$; $p(w) = v$; add w to L}}




f
d
e
c
a
b
s

Acyclic: topological
$O(m)$

Non neg: shortest first (min d) (Dijstra)
$O(n^2)$ original, $O(m \log n)$ standard heap
$O(n \log n + m)$ F-heaps

General: FIFO-scanning
queue = L
$O(nm)$
↓
$O(\sqrt{n}\, m \log C)$ (cost-scaling)

Dijkstra alg:

Monotonicity on heap:

v's are deleted from Q in increasing order by d!

Dial: (small integer edge wts) $O(m + C_n)$

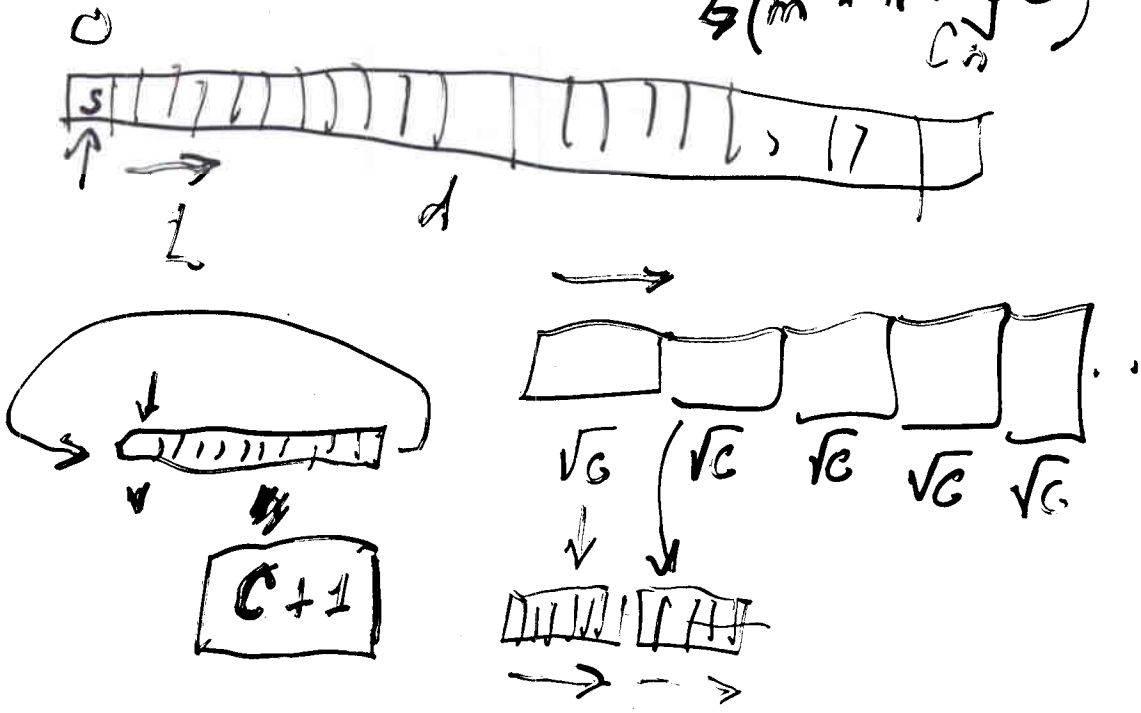
$\leq C$

$\leq$

$(m + 2\sqrt{C}n)$

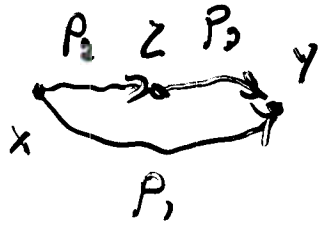
$\Theta(m + n \log C)$

C_n



All pairs:

Dynamic prog.



$d(x,x) = 0$

$d(x,y) = \infty$ ~~for all~~ if $x \neq y$, $(x,y) \notin E$

$d(x,y) = c(x,y)$ if $x \neq y$, $(x,y) \in E$

for z

for x

for y

if $d(x,z) + d(z,y) < d(x,y)$ then

$d(x,y) = d(x,z) + d(z,y)$

$O(n^3)$

n single sources:

→ Dijkstra: $O(nm + n^2 \log n)$

1 Bellman-Ford → eliminate neg edge costs

$p(v)$

$c'(v,w) = c(v,w) + p(v) - p(w) \geq 0$