CSE 101 5-21-20

PA6: extend 1 more day

continue ... SSSP in weighted graph.

<u>Lemma 1</u>

let $x \in V(G)$, suppose that after Initialize(G, s), some seq. of calls to Relax$(\cdot,\cdot)$ results in $d[x]$ being finite. Then G contains an s-x path of weight $d[x]$.

Lemma 2

After Initialize(G, s), the inequality

$$\delta(s,x) \leq d[x] \quad (\text{for all } x \in V(G))$$

is maintained over <u>any</u> sequence of calls to Relax$(\cdot, \cdot)$.

Lemma 3 (Path Relaxation Property)

If

$$P : s = x_0, x_1, x_2, \ldots, x_k$$

is a minimum weight s-x_k path, and the edges of P are relaxed in order

$$(x_0, x_1), (x_1, x_2), \ldots\ldots, (x_{k-1}, x_k),$$

then $d[x_k] = \delta(s, x_k)$. This is true regardless of any other relaxations that occur, even if they are interspersed with the above relaxations.

BellmanFord

see Examples/pseudo-code/GraphAlgorithms

Runtime: let $n = |V(G)|$, $m = |E(G)|$

1st loop costs $(n-1)m = \Theta(n \cdot m)$

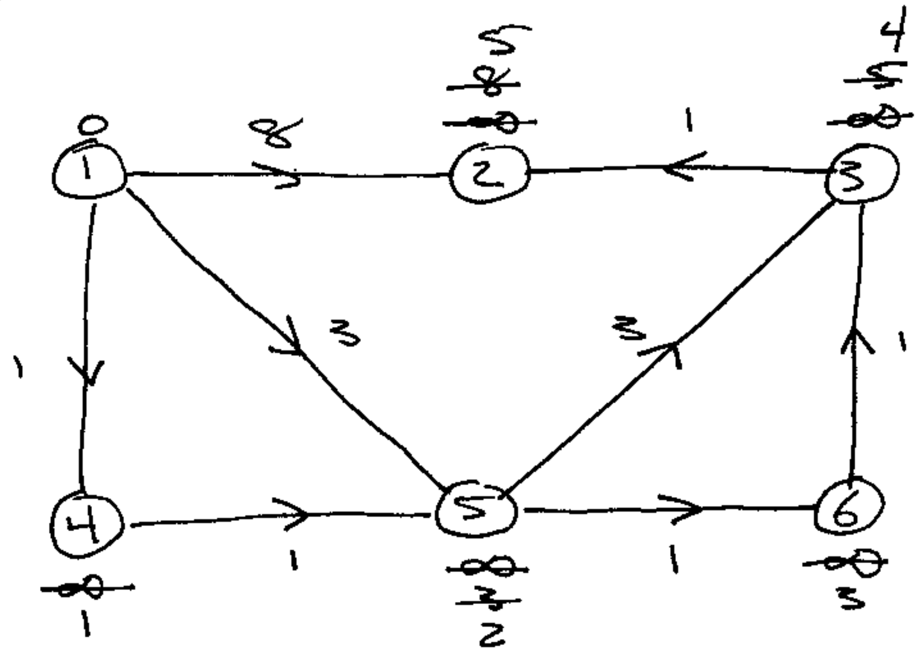
2nd loop costs $m = \Theta(m)$

total cost $= \Theta(nm)$.

Dijkstra

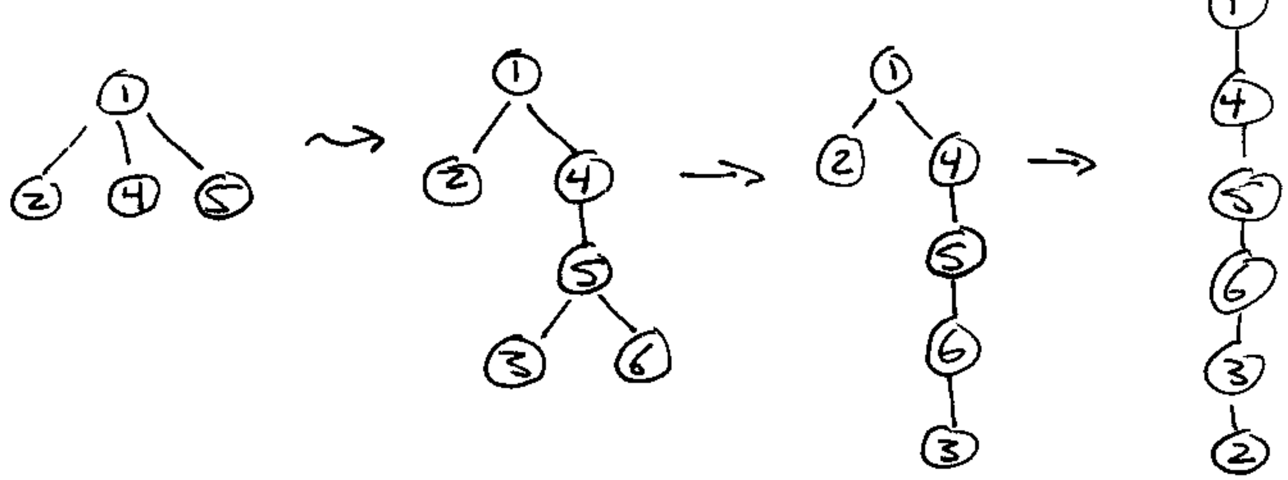
see /Examples/Pseudo-code/Graph Algorithms

Ex. $s = 1$



PQ: ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~

shortest Paths Tree:



Runtime :

total cost = $\Theta\left((n+m)\log(n)\right)$

Recall $\frac{\log(n)}{n} \xrightarrow[\text{as } n\to\infty]{} 0$

$\therefore \log(n) = o(n)$

<u>Remember</u> : Dijkstra <u>needs</u>
edge weights all non-negative.

Chapter 23: Minimum Weight Spanning Tree Problem (MWST)

assume G is an undirected graph.

Defn
let $G=(V,E)$ be a graph. A spanning tree in G is a subgraph T satisfying:

(1) T is a tree
(2) $V(T)=V(G)$

let $G=(V,E)$ be a weighted graph

$$w: E \rightarrow \mathbb{R}$$

Defn

let H be a subgraph of G.
Then the weight of H is

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

Defn

A minimum weight spanning tree in G is a sp. tree T such that

$$w(T) \leq w(S)$$

for all sp. trees S in G.

MWST Problem

determine a mwst in G.

Algorithms for MWST:

- Kruskal
- Dual-Kruskal
- Prim

Kruskal uses a min-Priority Queue whose records are edges, and whose keys are edge weights.

summary:

- start with ϕ
- from amongst all edges whose addition to current set would not create a cycle, add one of minimum weight

Note:

G contains a sp. tree iff

G is connected.

Kruskal(G)

$Q = E(G)$ // keys = edge weights

$F = \phi$

while $|F| < n-1$ // where $n = |V(G)|$

  $e = \text{extractMin}(Q)$

  if $F \cup \{e\}$ is acyclic

    $F = F \cup \{e\}$