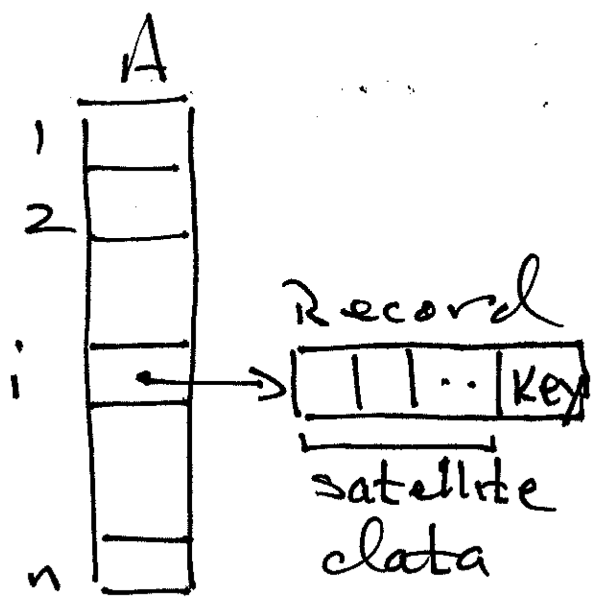
CS2 101 12-4-25

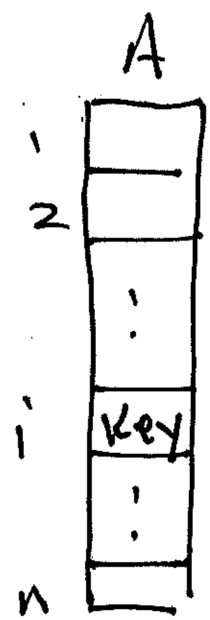
- Final Exam: Th 12/11 12:00-2:00 pm
- Pa8: Fri. 12/5 10:00 pm (last)
- SETs: Sun. 12/7 11:59 pm

Priority Queues (heap implementation)

General Picture



Our Picture



Exercise

re-write all algorithms in General Picture.

Chap. 24 : Graph algorithms

24.1 SSSP in a weighted graph

Definitions

- Weighted graph

A Graph $G=(V,E)$ with

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

- Path weight : $w(p)$

if $p : u = x_0, x_1, \ldots, x_k = v$, then

$$w(p) = \sum_{i=1}^{k} w(x_{i-1}, x_i)$$

- shortest Path Weight: $\delta(u,v)$

$$\delta(u,v) = \begin{cases} \min\{w(P) \mid P \text{ is a } u\text{-}v \text{ Path}\} & \text{if } v \text{ reachable from } u \\ \infty & \text{otherwise} \end{cases}$$

- Shortest Path:

  a Path P with $w(P) = \delta(u,v)$.

- SSSP:

  Given a weighted graph G and $s \in V(G)$ find.

  (1) $\delta(s,x)$ for all $x \in V(G)$

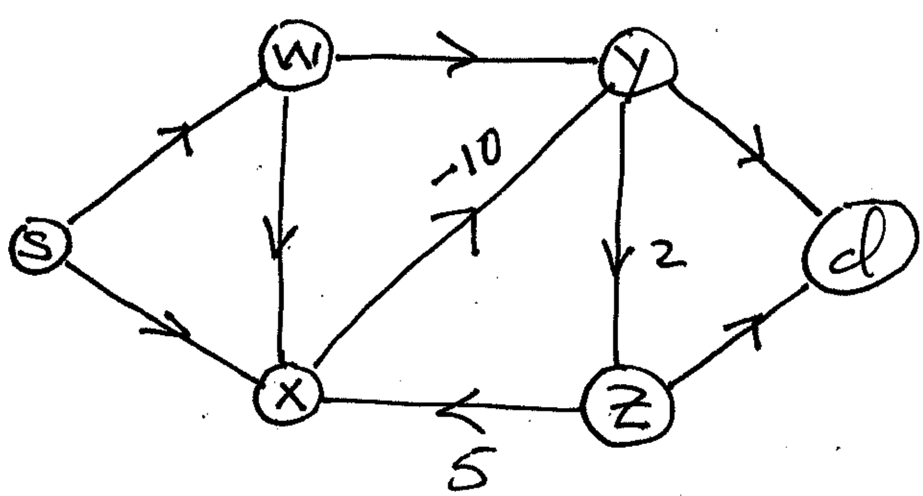
  (2) if $\delta(s,x) < \infty$, find a shortest s-x Path.

Two Algorithms

- BellmanFord
- Dijkstra

Both actually find a min-weight walk, which is necessarily a path, as long as there are no negative weight cycles reachable from s.

Ex



Cycle: (x, y, z, x) weight $= -3$

- Dijkstra:
  works only on graphs with non-negative edge weights.

- BellmanFord:
  allows neg. weight edges
  Returns false if a neg. weight cycle is reachable from s, true otherwise.

Vertex attributes

$P[x]$: Parent, Predecessor

$d[x]$: distance estimate $\delta(s,x)$

As in BFS: Predecessor Subgraph

$G(V_p, E_p)$.

$$V_p = \{x \in V \mid P[x] \neq nil \text{ or } x = s\}$$

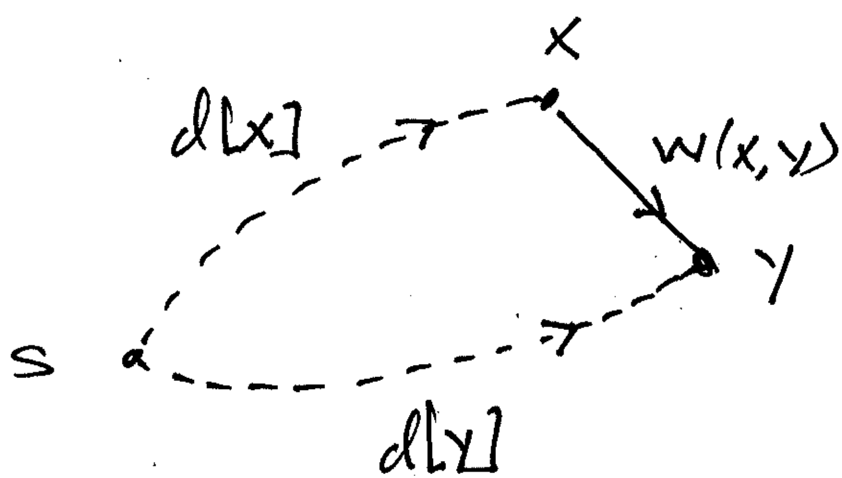
$$E_p = \{(P[x], x) \mid P[x] \neq nil\}$$

This is a Shortest Paths Tree.

Two subroutines :

Initialize(G, s) :

Relax(x, y) : pre: $y \in adj[x]$



note: • Relax(x, y) changes y only

• after Relax(x, y), then

$d[y] \leq d[x] + w(x,y)$

Lemma (Path relaxation Property)

If

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

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

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

then $d[x_k] = \delta(s, x_k)$. This is true regardless of any other relaxations that take place, even if interspersed with these relaxations.

Bellman Ford:

$$\text{cost} = \Theta(m \cdot n)$$

$$m = |E(G)|, \quad n = |V(G)|$$

Dijkstra

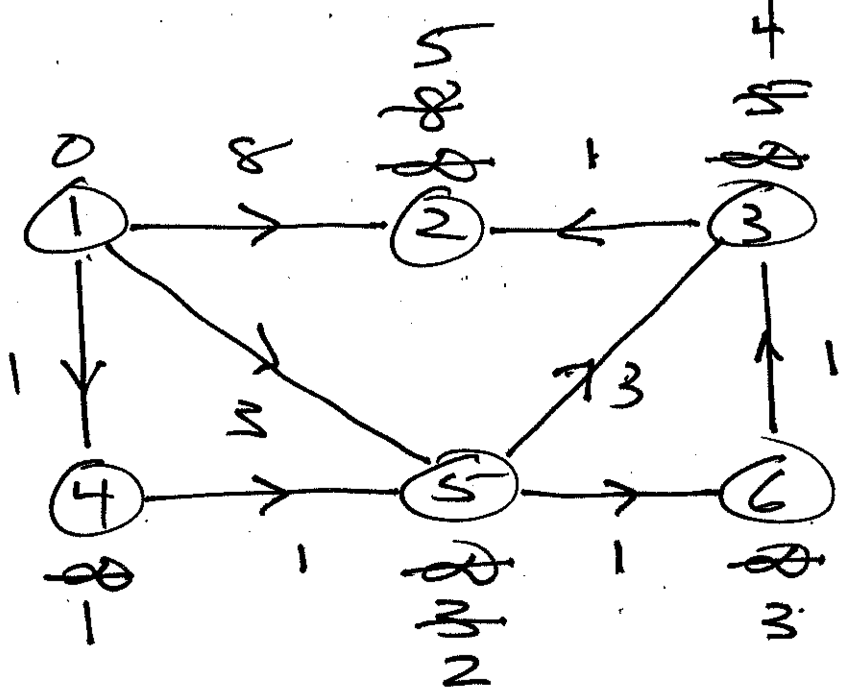
P.Q.

$$\text{cost} = \Theta((m+n)\log n)$$ with

or

$$\text{cost} = \Theta(m + n^2)$$ without

Ex. $s = 1$



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

d: 0 ~~8~~ 5 ~~5~~ 4 1 ~~3~~ 2 3

p: n ~~1~~ 3 ~~5~~ 6 1 ~~1~~ 4 5

Remove order: 1, 4, 5, 6, 3, 2