CSE 101 3-14-25

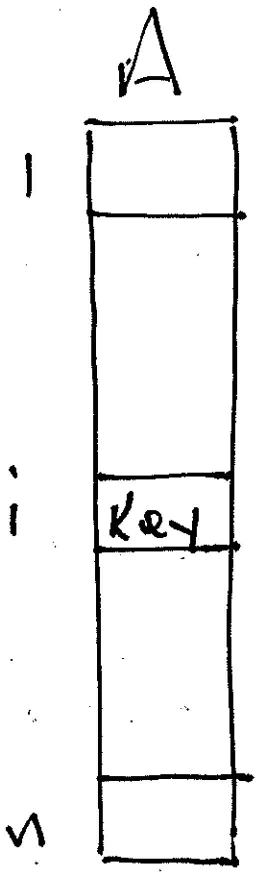
cost of HeapIncreaseKey()

$= \Theta(\log n)$

Exercise

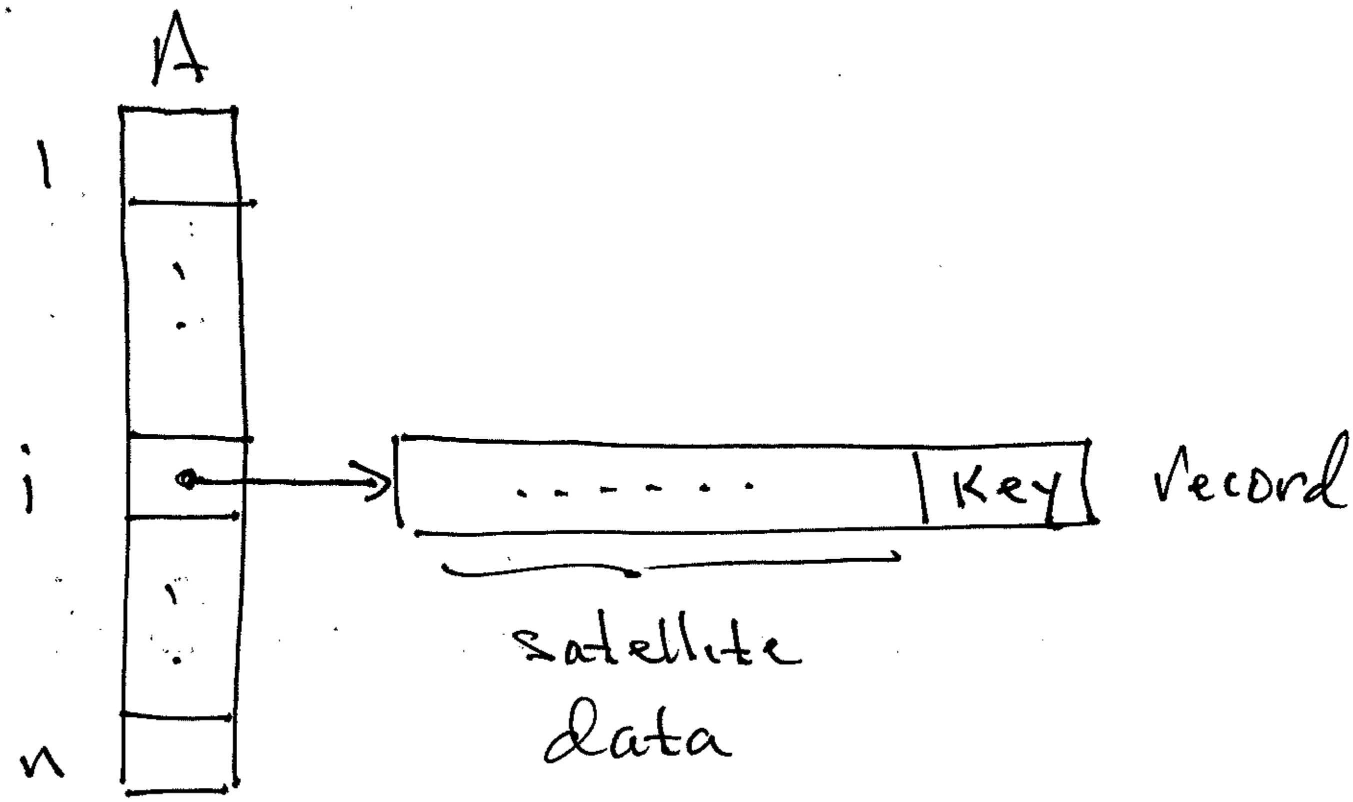
Write Min(), DeleteMin(), ExtractMin(), DecreaseKey(), Insert() for a min-P.Q.

Our Picture



no satellite data

General Picture



Exercise

re-write all Heap Pseudo code in general Picture.

Chap. 24

SSSP in Weighted graphs

Weighted graph $G = (V, E, w)$

$$w : E \to \mathbb{R}.$$

Path Weight

$$p : \underbrace{v_0, v_1}, \underbrace{v_1, v_2} \ldots, \underbrace{\ , v_k}$$

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

Shortest Path Weight

$$\delta(x,y) = \begin{cases} \text{weight of a min-weight } x\text{-}y \text{ path, if it exists} \\ \text{i.e. if } y \text{ reachable from } x \\ \infty \quad \text{otherwise} \end{cases}$$

Shortest x-y Path

Path p with $w(p) = \delta(x,y)$.

SSSP:

given $s \in V$, (1) determine $\delta(s,x)$ for all $x \in V$, and (2) for each x s.t. $\delta(s,x) < \infty$, find a shortest s–x path.

Two algorithms

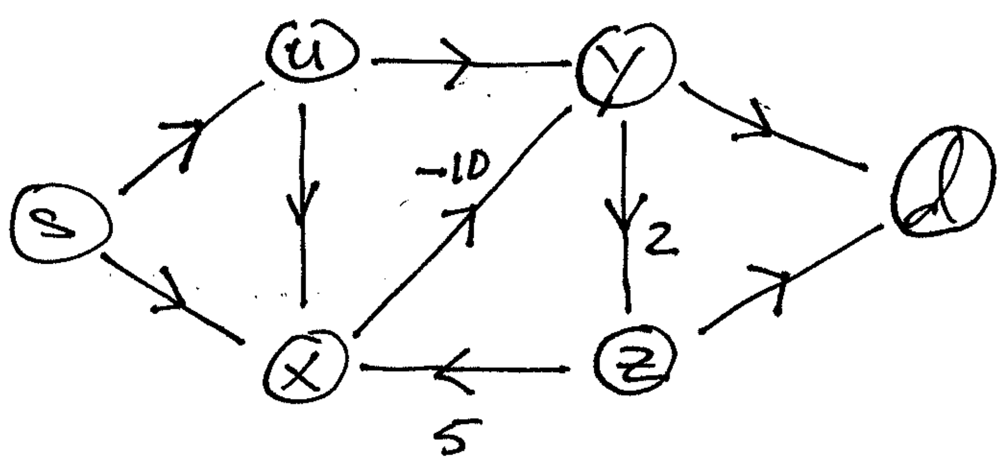
- Belman-Ford $\left\{ \begin{array}{l} \text{return true if} \\ \text{no neg. weight cycle} \\ \text{reachable from } s. \end{array} \right.$
- Dijkstra (restrict to only pos. edge weights)

Both actually find min-weight walks. note: this is automatically a min weight path.

Problem

Negative weight cycle reachable from the source.

Ex



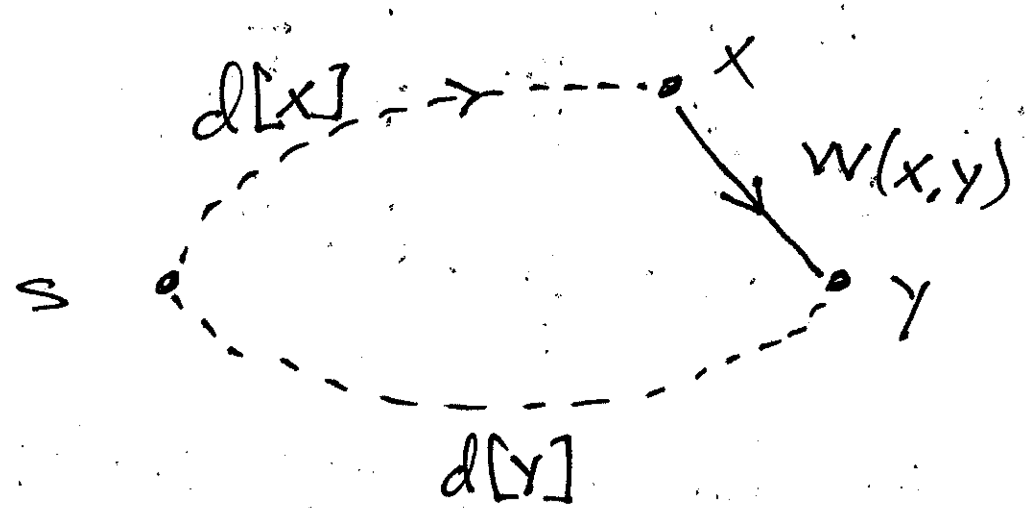
cycle: x, y, z, x weight = -3

As in BFS, we have

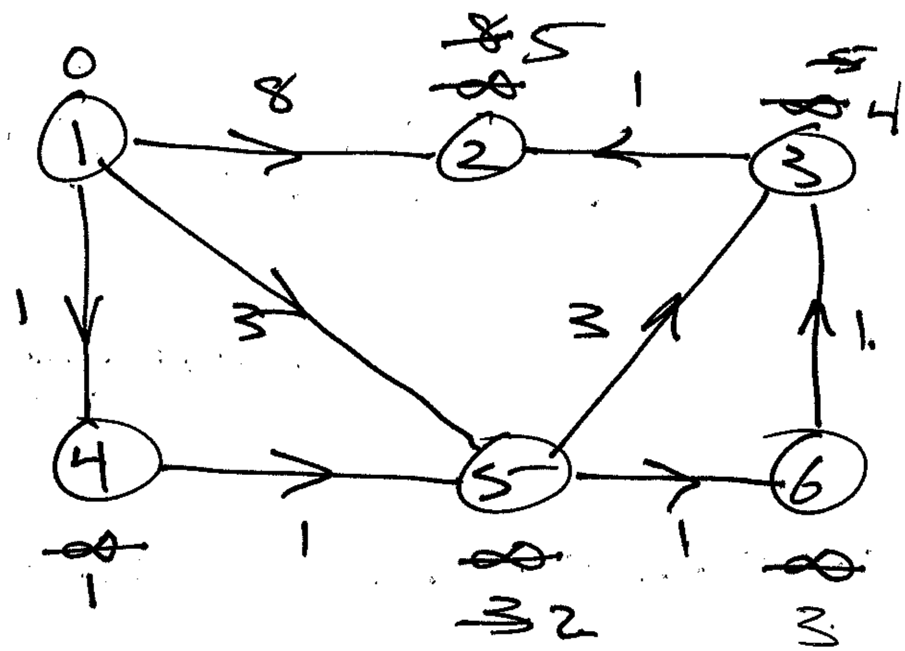
$P[x]$, $d[x]$, predecessor subgraph.

<u>Common Infrastructure</u>:

- Initialize(G, s)
- Relax(x, y)



Ex. Dijkstra : $s = 1$



extract

1

4

P.Q. :	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~
d :	0	~~8~~5	~~5~~4	1	~~3~~2	3
p :	nil	~~1~~3	~~5~~6	1	~~1~~4	5

Tree