CSE 101-01 6-8-23

- Final exam: Mon 6/12 12-2 PM
- SETs! Due Sun 6/11 11:59 PM
- PA8: one last day → Sat.

P.Q. operations using Heap $\begin{Bmatrix} \text{max} \\ \text{min} \end{Bmatrix}$

See pseudo-code: ✓

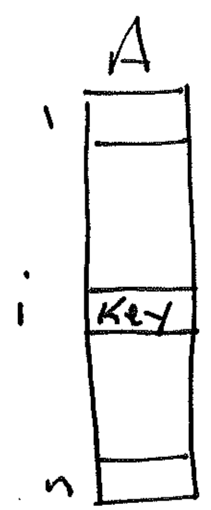
Runtimes:

HeapMaximum : $\Theta(1)$

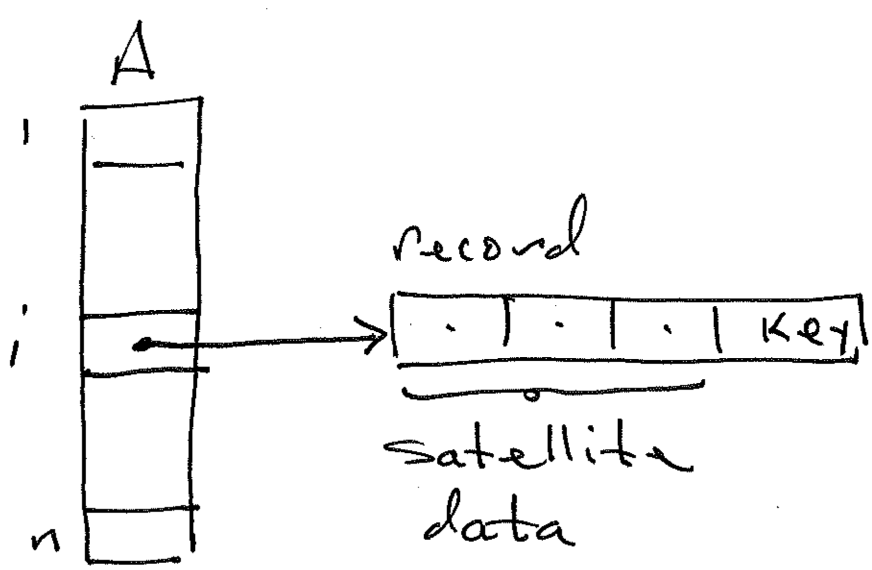
$$\left.\begin{array}{l} \text{" DeleteMax} \\ \text{" ExtractMax} \\ \text{" IncreaseKey} \\ \text{" Insert} \end{array}\right\} \Theta(\log n)$$

Heap Algorithms

our Picture:



General Picture:



Exercise: re-write algorithms in general Picture.

SSSP Problem in a Weighted graph

Ch. 24 of CLRS.

Defn

- a weighted graph is a graph $G=(V,E)$ with a fcn.

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

called weight fcn.

- weight of a path: $x = v_0, v_1, \ldots, v_k = y$ (P)

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

- Shortest Path Weight $x \to y$

$$\delta(x,y) = \begin{cases} \min\{w(p) \mid p \text{ is an } x-y \text{ Path}\} \\ \text{if such a Path exists} \\ \infty \text{ if no such Path exists.} \end{cases}$$

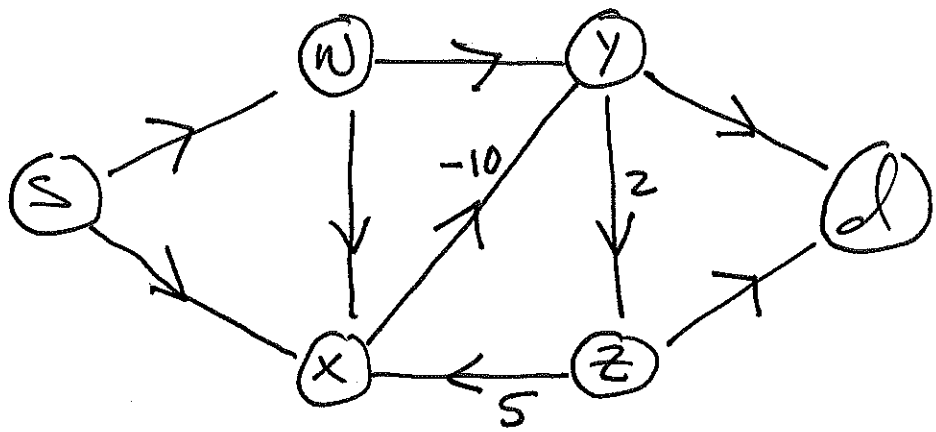
- a shortest $x-y$ Path is an $x-y$ path p s.t. $w(p) = \delta(x,y)$.

SSSP Problem

Given a weighted graph G, and a source $s \in V(G)$: (1) determine $\delta(s,x)$ for all $x \in V(G)$, and (2) for all x with $\delta(s,x) < \infty$, find a shortest $s-x$ Path.

Problem with SSSP:

Ex



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

Dijkstra: outlaws negative weight edges.

Infrastructure for Dijkstra:

for each vertex:

- $P[x]$: encodes a shortest path tree
- $d[x]$: estimate of $\delta(s,x)$

- Predecessor subgraph $G_p = (V_p, E_p)$

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

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

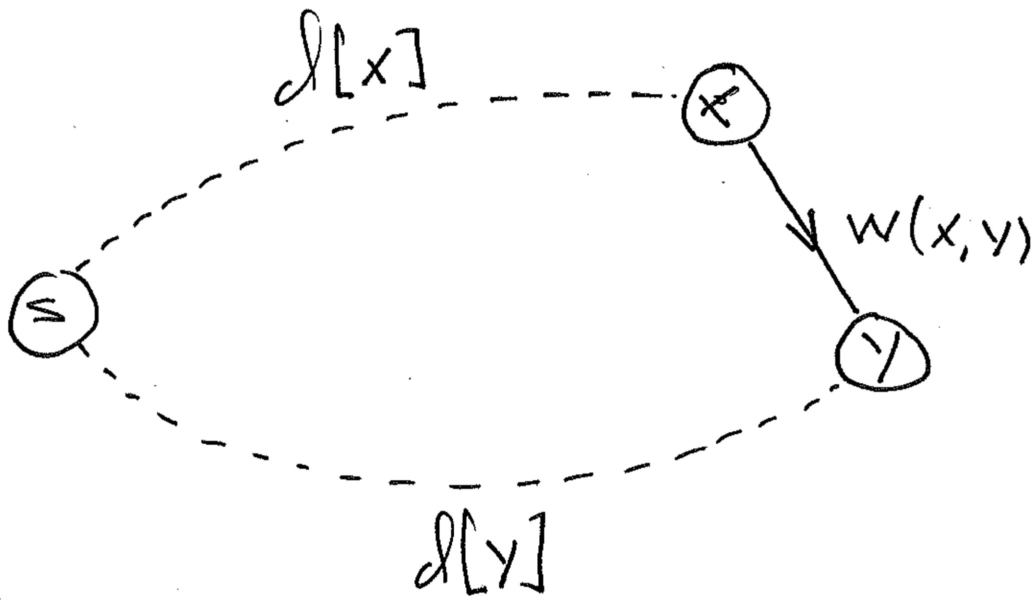
To print out a s.p. use

PrintPath(G, s, x)

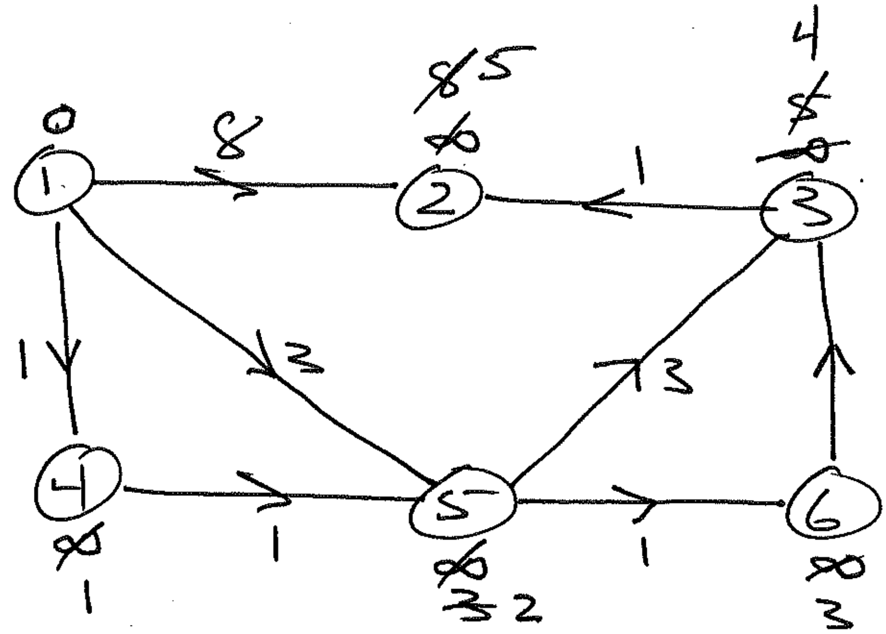
Two helper fcns.

- Initialize(G, s)

- Relax(x, y) Pre: $(x, y) \in E(G)$



Ex. $s = 1$

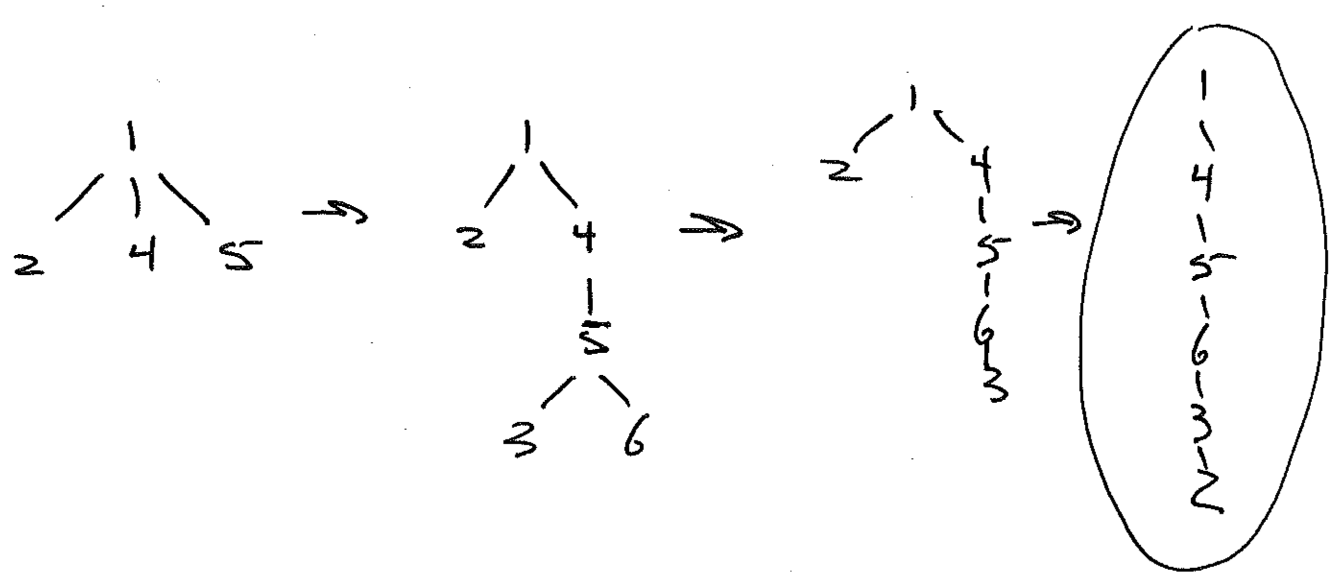


Extract order

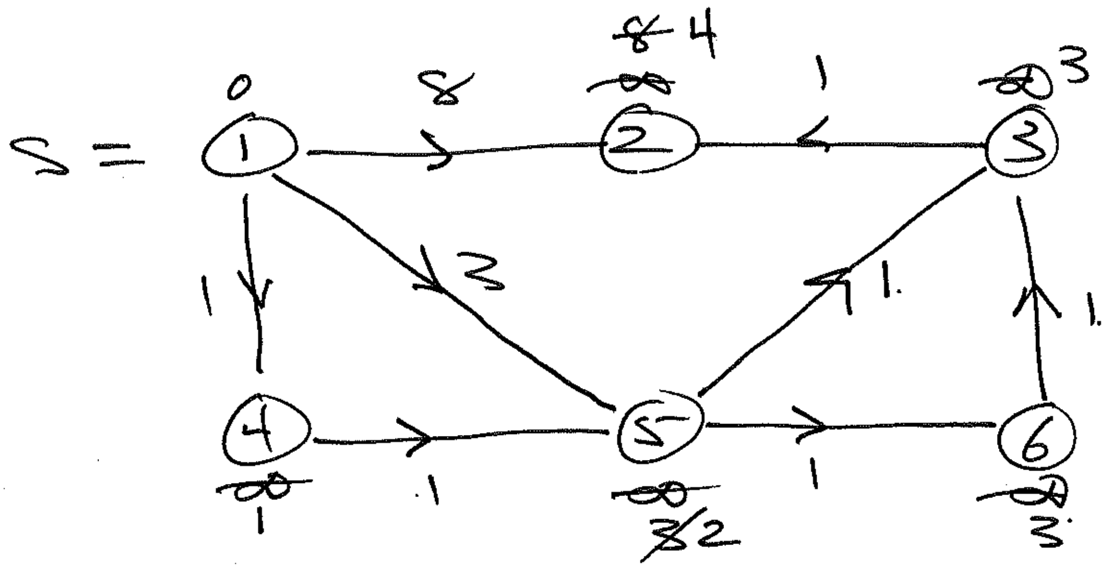
1
4
5
6
3
2

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

Tree:



Ex. if mult. id. keys, extract vertex with smallest label.



ext. ord

1
4
5
3
6

PQ:	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~6~~
d:	0	~~8~~4	3	1	~~3~~2	3
p:	n	~~1~~3	5	1	~~1~~4	5

Tree: