CSE 101 6-2-26

Final: Wed. 4-6 PM

SETS:

P.Q. operations in a max heap.

See Examples/...

HeapMax : $\Theta(1)$

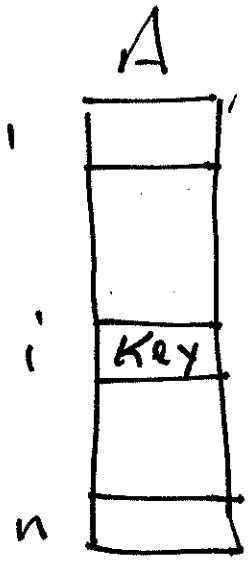
HeapDeleteMax : $\Theta(\log n)$

HeapExtractMax : $\Theta(\log n)$

HeapInsert : $\Theta(\log n)$

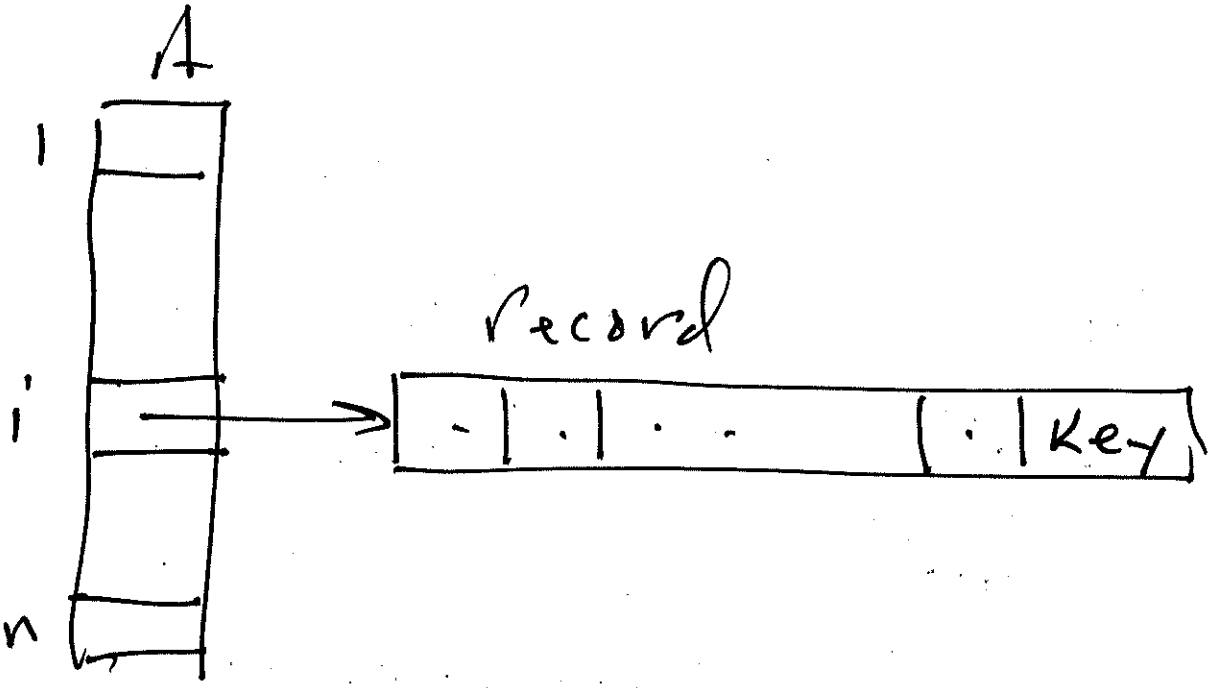
HeapIncreaseKey : $\Theta(\log n)$

our Picture of a Heap



no satellite data

General Picture of a heap



Exercise

- re-write all Pseudo code in general Picture.

- Implement min/max PQ in both C and C++.

chap 24: SSSP Problem in a Weighted graph.

Defn

- A weighted graph (directed or undirected) is a graph (V, E) with a function

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

We call $w(x,y)$ the weight of edge xy.

- The weight of a path p $u \overset{p}{\rightsquigarrow} v$

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

is

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

- Let $x, y \in V(G)$. The <u>shortest path weight</u> is

$$\delta(x,y) = \begin{cases} \min\{w(p) \mid p \text{ is a } x\text{-}y \text{ path}\} & \text{if } y \text{ is reachable from } x \\ \infty & \text{otherwise} \end{cases}$$

- A shortest x-y path in G is an x-y path p such that $w(p) = \delta(x,y)$. (Provided y is reachable from x)

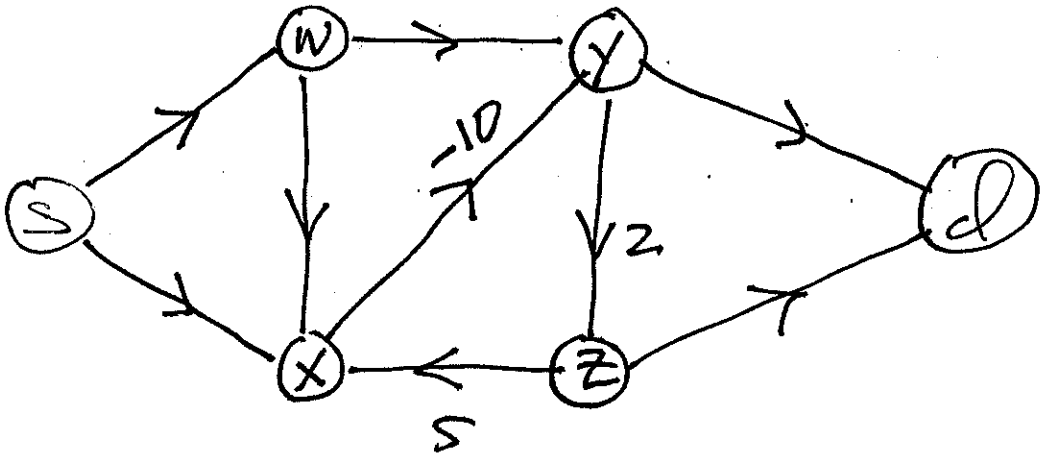
- SSSP problem:
  Given a source $s \in V(G)$
  (1) determine $\delta(s,x)$ for all $x \in V(G)$
  (2) if $\delta(s,x) < \infty$, find a shortest s-x path p.

Two Algorithms:

- Bellman-Ford { allows negative edge weights
- Dijkstra { outlaws negative edge weights.

Both actually a shortest $s-x$ walk (as opposed to a path).

Ex. negative weight cycle, reachable from source.



observe $w(x, y, z, x) = -3$

- Dijkstra: no neg. weight edges
- Bellman-Ford: Detects neg. weight cycles reachable from s, returns false in this case.

Common infrastructure:

$\pi[x]$: parent/predecessor, like in BFS.

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

Predecessor subgraph: (V_π, E_π)

$$V_\pi = \{x \in V \mid \pi[x] \neq nil\} \cup \{s\}$$

$$E_\pi = \{(\pi[x], x) \mid \pi[x] \neq nil\}$$

Functions

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

Picture for Relax