cse 101 6-1-21

- quizzes : today
- final : mon June 1 _ 12-8PM, 2hrs
- SETs : closes next Sunday.
- PA7 : ext 1 day → Friday.

Priority Queues (max)

- Implement using (max)-heap
- see pseudo-code

Runtimes: $n = \#$ keys

- HeapMaximum() : $\Theta(1)$
- HeapDeleteMax() : $\Theta(\log n)$
- HeapExtractMax() : $\Theta(\log n)$
- HeapIncreaseKey() : $\Theta(\log n)$
- HeapInsert() : $\Theta(\log n)$

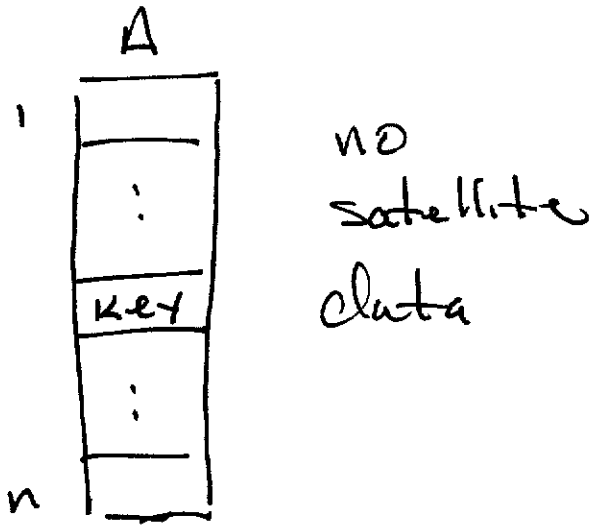
Exercise

re-write

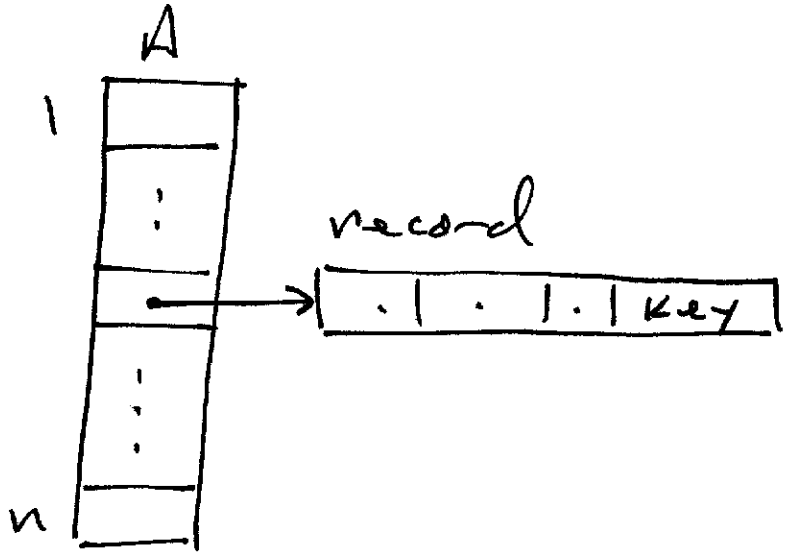
- heap operations
- PQ operations

for min case.

OUR Picture



General Picture



Exercise:

re-write PQ. operations in the general Picture.

Exercise

Implement a P.Q. (max, min, both) in C and C++.

Why have a Priority Queue?

SSSP in a Weighted Graph

Defn

A weighted graph (dir. or undir.) is a graph $G = (V, E)$ with a weight function

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

(weight, cost, distance) of an edge.

Defn

Let $P: x = v_0, v_1, \ldots, v_k = y$ be an x-y path in G. The **weight of P** is

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

Defn shortest path length

$$\delta(x,y) = \begin{cases} \min\{w(P): P \text{ is an } x\text{-}y \text{ path}\} \\ \quad \text{if } y \text{ reachable from } x. \\ \infty \text{ if } y \text{ not reachable from } x. \end{cases}$$

A **shortest x-y path** P is a path with $w(P) = \delta(x,y)$.

SSSP in a weighted graph:

Given $s \in V(G)$, determine $\delta(s,x)$ for all $x \in V(G)$. Further, if $\delta(s,x) < \infty$, then determine a shortest $s-x$ path in G.

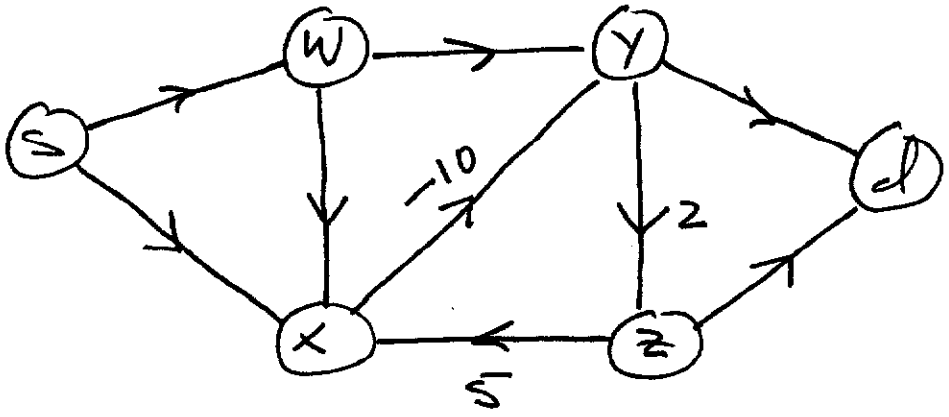
Two Algorithms:

- Dijkstra: requires positive weights
- Bellman-Ford: allows negative weights

Note: Both algorithms actually find shortest walks. ordinarily shortest walks are shortest paths.

Exception: negative weight cycle reachable from the source s.

Ex.



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

There is no min weight s-d walk!

- Dijkstra's strategy: outlaw neg. weight edges
- Bellman-Ford strategy: detect neg. weight cycles reachable from source, return False in that case.

New Graph Infrastructure

Vertex Attributes :

- $\pi[x]$: Parent (Predecessor), encodes shortest Path tree.
- $d[x]$: estimate of $\delta(s, x)$.

Predecessor subgraph :

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

(vertices reachable from s)

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

$$G_\pi = (V_\pi, E_\pi)$$

Helper functions:

Initialize(G, s)

1. for all $x \in V$
2. $\quad d[x] = \infty$
3. $\quad p[x] = nil$
4. $d[s] = 0$

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

1. if $d[y] > d[x] + w(x, y)$
2. $\quad d[y] = d[x] + w(x, y)$
3. $\quad p[y] = x$

Picture