Oct 25

$O(1)$ **Dijkstra's Algo**

$d'(w) = \min\limits_{\substack{u \in R \\ (u,w) \in E}} \{d(u) + l_{(u,}$

0. $R = \{s\}$, $d(s) = 0$

1. While $\exists\ x \notin R,\ u \in R$ s.t $(u,x) \in E$ $\leftarrow \leq n$ ti

(**) $\rightarrow$ Pick w that min $d'(w)$ [Among all such x's

Add w to R $\}\ O(1)$

$d(w) = d'(w)$

Def: Let P_u be the s-u path in "Dijkstra tree"

THEOREM: $\forall\ u \in V$, P_u is a shortest s-u path.

$\Longrightarrow$ $d(u)$ computed is correct $\Longrightarrow$ Dijkstra is correct

(Ex.)

LEMMA: At the end of each iteration, $\forall\ u \in R$; P_u is a shortest s-u path.

Ex: $u \in V$ s.t $\exists$ s-u path $\Longleftrightarrow$ $u \in R$ at the end of the algo

---

Lemma + Ex $\Longrightarrow$ THEOREM

Pf (idea) of lemma: By induction on $|R|$

Base case: $R = \{s\}$, $d(s) = 0$ ✓

Inductive hypothesis: Assume true for $|R| = k$ $(k \geq 1)$

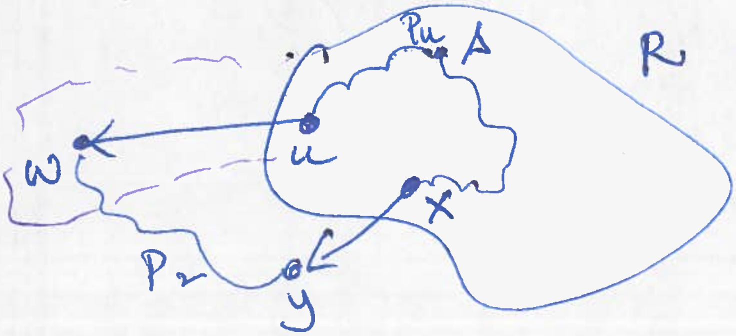
Inductive Step: (Argue for $|R| = k+1$)
Assume w is added to R at the end of iteration $k+1$.

$P_w = P_u, w$

Goal: Argue P_w is a shortest s-w path.

Assume for sake of contradiction, $\exists$ an s-w path P'_w s.t

$l(P'_w) < l(P_w)$ —— (*)



As $s \in R$ & $w \notin R \Rightarrow \exists\ x \in R, y \notin R$ s.t $(x,y) \in R$ where

$$P'_w = P_x, y, P_2$$

$$l(P'_w) = l(P_x) + l_{(x,y)} + l(P_2)$$

$$\text{I.H. on } x \rightarrow = d(x) + l_{(x,y)} + \underbrace{l(P_2)}_{\geq 0} \quad (\text{as } l_e \geq 0\ \forall e)$$

$$\geq d(x) + l_{(x,y)} \underset{\text{defn of } d'}{\geq} d'(y) \underset{\text{Algo defn}}{\geq} d'(w) = l(P_w)$$

$\Rightarrow l(P'_w) \geq l(P_w) \Rightarrow$ contradicts (*) ∎

Runtime Analysis of Dijkstra [Want to concentrate (**)]

→ Maintain an array R of length n

$$R[u] = \begin{cases} T & \text{if } u \in R \\ F & \text{o/w} \end{cases}$$

Init: $R[s] = T$, $R[u] = F$ $\forall u \neq s$ $\boxed{O(n)}$

→ Another array of length n, d

$d[u]$ = stores the final $d(u)$ value.

$O(n)$ Init: $d[s] = 0$, $d[u] = \infty$ $\forall u \neq s$

Implement (**)

Take 1: In every iteration, compute

$$d'(w) = \min_{\substack{u \in R \\ (u,w) \in R}} \{d(u) + l_{(u,w)}\}$$

$\forall w \in V$, look at all ~~u ∈ R s.t~~ $(u,w) \in E$

compute $d(u) + l_{(u,w)}$ & compute the min among all such u's.

$\Rightarrow O(|n_w|+1)$

Doing this for all $w \in V$, $\sum_u O(|n_w|+1) \leq O(m+n)$

Can compute w that min $d'(w)$ in $O(n)$ time

$\Rightarrow$ (**) in $O(m+n)$ time. $\Rightarrow$ OVERALL: $O(n(m+n))$

Replace by $\log n$

Idea: Use a priority Q to keep track of d' values.

[Recall: Priority Q n pairs (ID, value)

Init: $O(n)$, ExtractMin: $O(\log n)$ w $d'(w)$

ChangeValue (ID, v') : $O(\log n)$ time

$\Rightarrow$ To implement m ChangeValue ops & n ExtractMin

$\Rightarrow O(\log n(n+m))$ ∎