3 Data Structures & Algs (Prims + Kruskal) 2000 P4q6 1

MR p11a,5

<u>Prims</u> (Rather like Dijkstra's)

$G(V, E)$

$V = \{v_1, v_2 -- v_n\}$

$E = \{(v_i, v_j, k_{ij}) --- \}$

Let $S = \{v_1\}$

Find edge (v_i, v_j, k_{ij})

when $v_i \in S$ and k_{ij} is minimum.

$v_j \notin S$

$\Rightarrow v_j \leftarrow$ v_j is "closest" edge to S

$S \leftarrow S \cup \{v_j\}$

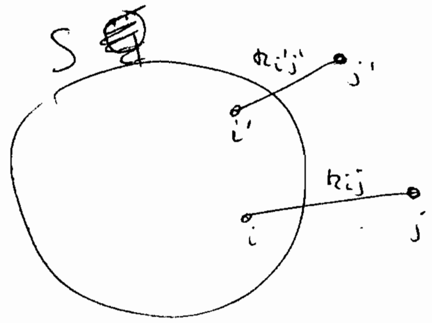
repeat until $S = V$.

Collection of edges forms the spanning tree.

<u>Proof of correctness</u>

If at some stage

an edge is chosen for which k_{ij} is not minimum



ie $k_{i'j'} < k_{ij}$

and edge $i'j'$ not used

edge ij used.

Delete ij from tree use $i'j'$ instead,

Total cost reduced by $(k_{ij} - k_{i'j'}) > 0$

and result still a spanning tree,

Kruskal

$G(V, E)$ as before

let $S_i = \{v_i\}$ $\forall i = 1 \ldots n$

find cheapest edge $(i\ j)$ join different sets S_p S_q

$S_p \Leftarrow S_p \cup S_q$

remove S_q

repeat until only one set remains.

The edges selected form ~~the~~ a minimum cost spanning tree.

Proof (almost identical to previous)

Cost of Prims

At each stage find minimum cost edge.

with binary heap.

Build heap cost $O(n)$

cost of extraction $O(\log_2 n)$

This is done $O(n)$ times

so ~~total~~ cost $O(n \log n)$

cost of looking at edges $O(|E| \log n)$

Total cost $O(n \log n + |E| \log n) = O(|E| \log n)$

Cost of Kruskal

Initials $O(n)$

Init heap $O(|E|)$

look at each edge once $O(|E| \times \log_2 |E|)$

same as Prims