

CISC 662 HW P Model Solution

ANONYMOUS@heaven.org

November 23, 2003

Exercise 23.2-4

Kruskal's Algorithm Kruskal's algorithm for finding MST can be broken into the following parts,

1. Make-Set takes $O(|V|)$ total
2. Sort edges ... takes $O(|E|\lg|E|)$ if no prior knowledge about weights.
3. Find-Set takes $O((|E| + |V|)\alpha(|V|))$.

Since we are given the information about edge weights, we can use counting sort or any linear sorting algorithm that uses prior knowledge of the input to achieve linear sorting, thus makes the time for sorting edges down to $O(|E|)$. Therefore, the dominant term becomes $O((|E| + |V|)\alpha(|V|))$. However, usually $|E| > |V|$, we can claim the whole thing to be:

$$O(|E|\alpha(|V|))$$

Since $\alpha()$ is bounded by $O(\log^*)$, we can also write the result to be:

$$O(|E|\log^* |V|)$$

given that \log^* is never over 5 for practical usage, we have:

$$O(|E|)$$

Grading Scheme:

sorting takes: $O(|E|)$ time (**Chap.8.2**)

union-find takes $O(|E|\log^* |V|)$ or $O(|E|\log^* W)$ time (with *union by rank* and *path compression*)

You'll get points as long as you stated the above. two points off if any one is missing, all points gone if none stated.

Exercise 23.2-5

Prim's Algorithm If implementing Priority queue using Fibonacci heap. \Rightarrow makes necessary operations (**Extract_Min**) into $O(\lg |V|)$. This cuts the running time down to $O(|E| + |V|\lg |V|)$.

In case where edge weight bounded by constant W , with linear sorting, the cost for single operation is reduced to a uniform $O(1)$, thus give the $O(|E| + |V|)$ total running time. However, $|E|$ usually dominates $|V|$, one can claim the final running time is $O(|E|)$. Thus we have:

$$O(|E|)$$

Grading Scheme

Extract_Min in Fibonacci Heap-based Priority Queue: $O(\lg |V|)$

Linear sorting cut cost per operation to $O(1)$ amortized if edge weight bounded by constant W .

Total running time be $O(|E| + |V|)$, $O(|E| + W)$, or $O(|E|)$

You'll get points as long as you stated the above. Two points off for not using Fibonacci heap, two points for linear sorting, one point for forgetting about total running time.