CS 5633 Analysis of Algorithms – Spring 08

4/3/08

9. Homework

Due **4/10/08** before class

1. **Union-Find (4 points)**

```
for(i=1; i<=16; i++) y[i]=MAKE-SET(x[i]);
for(i=4; i<=14; i+=2) UNION(y[i], y[i+1]);
for(i=3; i<=13; i+=5) UNION(y[i], y[i+3]);
UNION(y[7],y[11]); UNION(y[4],y[13]);
UNION(y[10],y[16]);
FIND-SET(y[6]); FIND-SET(y[5]);
```

   Assume an implementation of the Union-Find data structure with a disjoint-set forest with union-by-weight and path compression.

   Show the data structure after every line of code. What are the answers to the `FIND-SET` operations?

2. **Ackermann (2 points)**
   What is the value of $\alpha(10^{400})$? Justify your answer. (Remember, $\alpha(n)$ does not have a total upper bound as it does tend to infinity for large n, just very slowly.) What is the value of $\log_2(10^{400})$?

3. **Adjacency Matrix (6 points)**
   Suppose the graph $G = (V, E)$ is given in an adjacency matrix. The edge weights are the entries in the matrix.

   (a) How fast does DFS run on this graph? Justify your answer.

   (b) How fast does Prim's algorithm run on this graph? Justify your answer.

4. **Adding an edge (5 points)**
   Let $G = (V, E)$ be a connected undirected graph with weight function $w : E \to \mathbb{R}_0^+$ (i.e., all edge weights are ≥ 0). Assume edge weights are distinct. Further, let a minimum spanning tree T on G be given.

   Now, assume that one new edge (u, v), with $u, v \in V$, with weight $w(u, v)$ is added to G. (This weight is different from all other edge weights.)

   Give an efficient algorithm to test if T remains the minimum spanning tree for this new graph. Your algorithm should run in $O(|E|)$ time. Can you make it run in $O(|V|)$ time?

5. **Faster MST (4 points)**
   Let $G = (V, E)$ be a connected undirected graph with edge weights $w : E \to \mathbb{R}$.

   If all of the edge weights are integers between 1 and $|V|$, how fast can the minimum spanning tree be computed? (Give the *most efficient* algorithm you can think of.)

FLIP OVER TO BACK PAGE $\Longrightarrow$

6. **Kruskal's tree (4 points)**
   Since Kruskal's algorithm computes a forest of trees which are incrementally merged, it is not immediately obvious why the output is actually a tree.

   Argue why Kruskal's algorithm computes a tree. *(Hint: A tree is a connected acyclic graph.)* Do NOT use the fact that somebody else has proven that Kruskal's algorithm computes an MST.