To make no mistakes is not in the power of man; but from their errors and mistakes the wise and good learn wisdom for the future. – Plutarch

# FINAL EXAM

Total Score: 155. Do all problems. Problems marked (*) are more difficult but still part of the exam. Send exam to: jj2903@nyu.edu

1. (15) In a Binary Search Tree $T$ define $desc[v]$ to be the number of descendents (including $v$ itself) of $v$.
   (a) (5) Suppose $v$ has children $y, z$ with $desc[y] = 10$ and $desc[z] = 20$. What is $desc[v]$?
   (b) (10) (*) Give a recursive program $HAOBO[v]$ that finds $desc[w]$ for *all* descendents $w$ of $v$, including $v$ itself. (Idea: Modify In-Order-Tree-Walk) When $v = Root[T]$ your program should take time $O(V)$.

2. (20) Suppose in $DFS[G]$ that $d[v] = 30$ and $f[v] = 50$.
   (a) (10) Suppose $w$ is Grey at time 30. What are its possible colors at time 50? Give full argument for your answer!
   (b) (10) Suppose $w$ is White at time 30. What are its possible colors at time 50? Give full argument for your answer!

3. (15) Consider the following program with input $M$

```
1. FOR S = 1 TO M
2. TEMP = S
3. WHILE TEMP ≤ M
4. TEMP = TEMP + TEMP
5. ENDWHILE
6. ENDFOR
```

   (a) (5) For a given $S, M$ how many times do we hit step 3?
   (b) (5) Write as a sum the total number times we hit step 3?
   (c) (5) (*) Evaluate the above sum as $\Theta(g(N))$ for some nice function $g(N)$ – analysis please!

4. (5) Which is faster (or are they both the same) when $n$ is large, a $\Theta(n^2)$ algorithm or a $\Theta(n^{3/2} \lg^2 n)$? (*Brief* reason, please.)

5. (15) Consider Kruskal's Algorithm for MST on $V = 1000$ vertices and $E = 5000$ edges.

   (a) (5) Let $w$ be a vertex. How many different values can $\pi[w]$ have during the course of the algorithm?

   (b) (5) Suppose at some point in the algorithm that $x_0, \ldots, x_L$ are such that $\pi(x_i) = x_{i+1}$ for $0 \leq i < L$ and $SIZE[x_L] = 50$. What is the maximal possible value of $L$?

   (c) (5) (*) Let $v$ be a vertex. What is the maximal number of possible values of $SIZE[v]$ in the course of the algorithm?

6. (20) `Short Stuff:` Brief answers – no arguments needed.

   (a) (5) Give an exciting result – one discussed in this class – that was found less than thirty years ago.

   (b) (5) What is the quickest way (worst case) to sort a million non-negative integers, all less than a trillion?

   (c) (5) When is the third smallest edge (assume no ties) *not* accepted in Kruskal's algorithm? (A picture will help!)

   (d) (5) Give an algorithm – one discussed in this class – that makes use of the heap (max or min) data structure.

7. (15) Consider the recursion $T(n) = 8T(n/2) + n^2$ with initial value $T(1) = 5$. (To avoid fractions, restrict to $n$ a power of two.)

   (a) (5) Using the Master Theorem find $T(n) = \Theta(g(n))$ for some nice $g(n)$.

   (b) (5) Setting $S(n) = T(n)/n^3$ give the recursion (including initial value) for $S$.

   (c) (5) (*) Show $T(n) \sim cg(n)$, $g(n)$ from the first part, for some *explicit* constant $c$.

8. (15) Apply the Extended Euclidean Algorithm to find $d = \gcd(15, 24)$ and $x, y$ with $15x + 24y = d$. *Show all work,* the answers alone will not suffice!

9. (10) Let $GOLDBACH$ be the set of integers expressible as the sum of two (not necessarily distinct) odd primes. For example, $18 = 11 + 7 \in GOLDBACH$. Show $GOLDBACH \in NP$. (Give clearly the roles of Oracle and Verifier.)

10. (15) Consider Prim's Algorithm (for MST) on a connected graph $G$ with $V$ vertices, $E$ edges, and designated source vertex $s$. *Assume* that $s$ is joined to *all* other vertices by an edge. *Further assume* that for every vertex $x \neq s$, amongst all edges using $x$ the edge $\{s, x\}$ has the smallest weight.

    (a) (5) Argue that the MST will consist of the $V - 1$ edges $\{s, x\}$, $x \neq s$.

    (b) (10) (*) Suppose further that $G$ is the complete graph. How long will Prim's Algorithm take *under these special assumptions* when $G$ is the complete graph – i.e., consists of *all* edges $\{x, y\}$.

11. (10) Let $G$ be a directed graph with designated source vertex $v$. Let $z \in G$, $z \in Adj[v]$, and *assume* the weight of $(v, z)$ is the smallest (assume no ties) of all the weights of $(v, y)$, $y \in Adj[v]$. Prove (yes, prove, using the algorithm is not a proof!) that the lowest weight path from $v$ to $z$ is given by the edge $(v, z)$. (A good picture will help!)

Do I contradict myself?
Very well then I contradict myself.
(I am large, I contain multitudes.)
– Walt Whitman