CS2223

FINAL EXAM

Name______________

Date: December 18, 2003
All documentation permitted

1	________
2	________
3	________
TOTAL	________

1. (35 points) Consider the following divide-&-conquer algorithm to find a minimum spanning tree of graph $G = (V, E)$ with weight function $w: E \to \mathbb{R}^+$.

Divide V into (essentially) equal sized sets V_1, V_2, where $|V_1| = \lfloor |V|/2 \rfloor$, $|V_2| = \lceil |V|/2 \rceil$.

Divide E into sets E_1, E_2, E_3 where E_1 is the edges of E with both endpoints in V_1

E_2 is the edges of E with both endpoints in V_2

E_3 is the edges of E with one endpoints in V_1 and one in V_2

That is, $E_1 = E \cap (V_1 \times V_1)$, $E_2 = E \cap (V_2 \times V_2)$ and $E_3 = E \cap (V_1 \times V_2)$

Let E_1^* be the edges of a minimum spanning tree of (V_1, E_1).

Let E_2^* be the edges of a minimum spanning tree of (V_2, E_2).

Let e be a shortest edge of E_3.

If $G = (V, E)$ is connected, must $E_1^* \cup E_2^* \cup \{e\}$ be the edges of a minimum spanning tree of G? Justify your response.

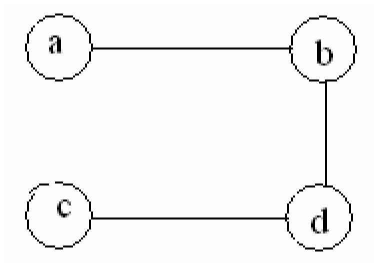
2. (35 points) Graph $G = (V, E)$ is a *tree* if it is connected and $|E| = |V| - 1$. Describe an algorithm to test if $G = (V, E)$ is a tree. Your algorithm must execute in time in $O(|V|)$

3. (30 points) Given an array $A[1..n]$ of non-negative reals, we want to find $\max_{1\le i\le j\le n}\left(\prod_{i\le k\le j} A[k]\right) = \max_{1\le i\le j\le n}\left(A[i]*A[i+1]*\ldots*A[j]\right)$. That is, we want to find the maximum product realizable as the product of a contiguous subsequence of numbers of A. For example, if A=(0.6, 0, 23, 0.5, 28, 0.9), then the answer is 322=23*0.5*28. If A=(0.8, 0.9, 0.6), then the answer is 0.9. If A=(1, 0, 12, 3, 1, 2, 0.001, 39), then the answer is 72=12*3*1*2. Describe an $O(n)$ algorithm to solve this problem.

CS2223

Solutions to Final Exam

1. The algorithm fails for the graph



if V is partitioned into $V_1 = \{a, c\}$ and $V_2 = \{b, d\}$.

2. We do a depth-first search on G, halting as soon as a backedge is encountered (G admits a cycle if and only if the dfs-forest has a backedge). The dfs-forest has $O(|V|)$ tree edges, and quits upon encountering a backedge, so the execution time is in $O(|V|)$.

3. *best_to*[1] ← $A[1]$

for $k \leftarrow 2$ **to** n **do**

best_to[k] ← max(A[k], *best_to*[k-1]*A[k])

and a (linear time) pass through *best_to* finds and returns its maximum value.