CS616 – Algorithms

Midterm Examination – Fall 2014
University of Virginia

Gabriel Robins

• This is a 6-hour take-home open-book, open notes, pledged exam.
• Note: while for your convenience the “possession time” of this exam is up to 24 hours, the actual “work time” to complete this exam should not exceed 6 contiguous hours.
• No collaborations, Web searches, nor communications with others are allowed during the exam.
• Do as many of the problems as you can; please explain/prove all answers.
• Shorter algorithms / explanations / proofs are much preferable to longer ones.
• Clearly state the short algorithm / proof idea first, and then your complete algorithm / proof.
• Clearly state the time complexity at the top of the page for each algorithm.
• Submit only the pages provided (use more sheets only if absolutely necessary).
• Derive answers on scratch paper first, then copy them neatly onto these pages.

During the exam, please feel free to ask clarifying questions using Email; responses will be posted to the class Web page (so please look at the class Web page often during this exam).

When you are done with this exam, please slip it under my office door (406 Rice Hall).

Good Luck!

Name:______________________________________________

Problem 1: 20 ________
Problem 2: 20 ________
Problem 3: 20 ________
Problem 4: 20 ________
Problem 5: 20 ________
Problem 6: 20 ________
Problem 7: 20 ________

Total: 140 ____________

“I think you should be more explicit here in step two.”
1) Give a time-\textbf{optimal} algorithm to determine the diameter of a weighted tree (i.e., longest path between any two nodes). Prove the optimality of your algorithm. What is the time complexity?

\textbf{Time complexity:}

\textbf{Proof of time-optimality:}

\textbf{Algorithm idea:}

\textbf{Algorithm details:}

**Algorithm idea:**

**Correctness proof:**

**Expected time complexity (as a function of p):**
Devise a time-optimal algorithm that given an arbitrary N-degree one-variable polynomial \( P \) with rational coefficients, and an arbitrary input value \( x \), computes \( P(x) \). Prove the optimality (as a function of \( N \)) of your algorithm. What is the time complexity?

**Time complexity:**

**Proof of time-optimality:**

**Algorithm idea:**

**Algorithm details:**
4) Solve problem 15.4-6 on page 397 in the [Cormen, 2009] textbook.

**Algorithm idea:**

**Algorithm details:**

**Time complexity:**

**Algorithm idea:**

**Algorithm details:**
6) The shortest (least total weight) path between two nodes in a weighted graph may be not unique. Give an algorithm to find a shortest path between two given nodes, which also minimizes the number of edges as a secondary criterion. What is the time complexity?

**Time complexity:**

**Algorithm idea:**

**Algorithm details:**
7a) Devise an efficient algorithm that given N points in the plane, determines a maximum (largest cardinality) collinear subset. What is the time complexity as a function of N?

**Time complexity:**

**Algorithm idea:**

**Algorithm details:**
7b) Devise an efficient algorithm that given N points in the plane, determines whether any three of the points are collinear and equally-spaced (along their containing line). What is the time complexity?

**Time complexity:**

**Algorithm idea:**

**Algorithm details:**

"Once you eliminate the impossible, whatever remains, no matter how improbable, must be the truth."
- Sherlock Holmes (by Sir Arthur Conan Doyle, 1859-1930)