Algorithms Qualifying Exam, August 2011

Computer Science Department

Michigan Technological University

Instructions:

1. This is a closed-book exam. You are not allowed to use the textbook, your class notes or lectures.
2. Please solve 5 problems out of 6. All problems are equally weighted. Please read all problems carefully and start with the problems you think are easier to solve.
3. You have 3 hours for this exam.
4. Please be precise and concise in your writing. It is important that you clearly present your solutions.
5. Unless a problem explicitly asks, you may reuse well-known algorithms or theorems without their proof of correctness.
6. For all problems that ask for the design of an algorithm, you should prove the correctness of your algorithms. Please, make your proof explicit in your solution.

1. Binary search of a sorted array takes logarithm search time, but the time to insert a new element is linear in the size of the array. We can improve the time for insertion by keeping several sorted arrays. Specifically, suppose that we wish to support SEARCH and INSERT on a set of n elements. Let $k = \lceil log(n+1) \rceil$, and the binary representation of n be $b_{k-1}b_{k-2}...b_0$. We have k sorted arrays A_0, A_1,...,A_{k-1}, where for i=0, 1, ..., k-1, the length of array A_i is 2^i. Each array is either full or empty, depending on whether $b_i = 1$ or $b_i = 0$, respectively. The total number of elements held in all k arrays is therefore $\sum_{i=0}^{k-1} b_i 2^i = n$. Although each individual array is sorted, there is no particular relationship between elements in different arrays.

   (a) Describe how to perform the SEARCH operation for this data structure. Analyze its worst-case running time. Your algorithm must take $o(n)$ time. **(10 points)**

   (b) Describe how to insert a new element into this data structure such that the amortized cost is better than $O(n)$. **(10 points)**

   You need to show that your algorithms satisfy the cost requirement.

2. Let $X = \{x_1, x_2, ..., x_n\}$ be a sequence of arbitrary real numbers. You are to design an algorithm to find the subsequence of consecutive elements x_i, x_{i+1}, ..., x_j whose product is maximum over all consecutive subsequences. The product of the empty subsequence is defined as 1. Observe that X can contain reals less than 1, including negative numbers.

   (a) Design an $O(nlogn)$ divide-and-conquer algorithm. **(10 points)**

   (b) Design a linear time dynamic programming algorithm. **(10 points)**

   Your answer to each subproblem should include the algorithm, proof of correctness, and analysis of complexity.

3. Consider the following three definitions in a connected undirected graph $G =< V, E >$, where V denotes the set of vertices of G, and E represents the set of edges of G:

   - **Minimum Vertex Cover** is a subset V' of V (i.e., $V' \subseteq V$) with minimum size such that every edge (u, v) in E has at least one end in V'.
   - **Maximum Clique** is a *complete* subgraph of G that has a maximum size.
   - **Maximum Independent Set** is a subset V' of V (i.e., $V' \subseteq V$) with maximum size such that for all u and v in V', edge (u, v) is not in E.

   (a) Explain how these problems relate with each other. **(10 points)**

   (b) Can we prove the NP-hardness of one using another one? Justify your answers. If your answer is affirmative, then present an example of a polynomial-time reduction from one of the above problems to another one. Otherwise, prove your claim. **(10 points)**

4. A Hamiltonian Cycle (HamCycle) in a connected graph is a cycle that visits every vertex of a graph exactly once. The HamCycle problem is in the class of NP-complete problems. Assume that we have an oracle that, given a graph $G = (V, E)$, decides in unit time whether or not G has a Hamiltonian cycle.

(a) Use such an oracle to design a polynomial-time algorithm that finds a HamCycle in a given graph $G = (V, E)$, where V denotes the set of vertices and E represents the set of edges of G. **(10 points)**

(b) Prove the correctness of your algorithm. Please explicitly mention what proof technique you use. **(10 points)**

5. In the Half 3-CNF Satisfiability problem, we are given a 3-CNF (3 Conjunctive Normal Form) formula ψ with n variables and m clauses where m is even. We wish to determine whether there exists a truth assignment to the variables of ψ such that exactly half the clauses evaluate to true and exactly half the clauses evaluate to false. Prove that HALF 3-CNF is NP-Complete. **(20 points)**

6. Design a *liner-time* algorithm that finds the minimum vertex cover of an undirected tree. Your answer should include:

   (a) an *algorithm* that finds the minimum vertex cover, **(8 points)**

   (b) a *proof* arguing that your algorithm is correct, **(6 points)**

   (c) and a *complexity analysis* that shows your algorithm is linear. **(6 points)**