長庚大學 105 學年度第一學期電機所博士班演算法資格考

1. Please write down your student ID and name on the answer sheet.
2. Please indicate the number of each your answer that is relative to the problem.
3. Any form of cheating will lead to fail.

Please select five problems to answer. Total score of this exam is 100. Maximum deduction of 20 points for each problem that your answer.

1. Given a list of n integers, *v*1, . . . , *v*n, the *product-sum* is the largest sum that can be formed by multiplying adjacent elements in the list. Each element can be matched with at most one of its neighbors. For example, given the list 1, 2, 3, 1 the product sum is $8 = 1 + (2 \times 3) + 1$, and given the list 2, 2, 1, 3, 2, 1, 2, 2, 1, 2 the product sum is $19 = (2 \times 2) + 1 + (3 \times 2) + 1 + (2 \times 2) + 1 + 2$.
   (a) Compute the product-sum of 1, 4, 3, 2, 3, 4, 2
   (b) Give the optimization formula for computing the product-sum of the first *j* elements
   (c) Give a dynamic program for computing the value of the product sum of a list of integers.
2. Consider the Hamiltonian Cycle problem: "Given a graph G, does G have a simple cycle containing all of the vertices of G?" (Recall that a simple cycle contains no edge or vertex twice) This problem is NP-complete because it can be reduced from Vertex-Cover which can be reduced from 3-Satisfiability. Now consider each of the following modifications of Hamiltonian Cycle. Do you think it is NP-complete or in P? If you believe it is in P then give a brief description of a method that will answer the question in polynomial time. If you believe it is NP-complete then give a brief justification such as an NP-complete problem that reduces to this problem
   (a) Hamiltonian Cycle restricted to trees
   (b) "Given a graph G and a list of k cycles in G, is one of these cycles a Hamiltonian Cycle?"
   (c) The Travelling Salesman Problem: "Given a graph G with weighted edges, what is the simple cycle of G containing all of the vertices of G that has maximum weight (if any such cycle exists)?"
3. Please describe any algorithm you know that use dynamic programming to solve all pairs shortest paths problem. And please describe the time complexity need for the algorithm you answer.
4. (a) Two of the most common divide-and-conquer sorting algorithms are quicksort and mergesort. In practice quicksort is often used for sorting data in main storage rather than mergesort. Give a reason why quicksort is likely to be the preferred sorting algorithm for this application.; (b) Quicksort's worst-case running time is $O(n^2)$, but it has an expected running time of $O(n\log n)$ if the partition function works well. What needs to be true about the partition function in order for the running time to be $O(n\log n)$? In practice, how can we ensure that this happens?
5. Suppose that you are given a graph G = (V, E) and a its minimum spanning tree T. Suppose that we delete from G, one of the edges $(u, v) \in T$ and let G' denote this new graph.
   (a) Is G' guaranteed to have a minimum spanning tree?
   (b) Assuming that G' has a minimum spanning tree T'. Is the following description true or false: the number of edges in T' is no greater than the number of edge in T'? Please explain your answer.
   (c) Assuming that G' has a minimum spanning tree T', describe an algorithm for finding T'. What is the work of your algorithm?
6. Answer the following question regarding Kruskal's algorithm for finding the minimum spanning tree
   (a) Describe briefly Kruskal's algorithm
   (b) Assume procedures FIND(*u*) and UNION(*u, v*) are given. Write the pseudo-code for Kruskal's algorithm
   (c) Write pseudo-codes for both FIND(*u*) and UNION(*u, v*) which run in $O(\log n)$
   (d) Prove the correctness of the algorithm