長庚大學 104 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 directed graph where each edge has a length, describe an algorithm that takes as input two vertices u, v and an integer $k \geq 0$ and outputs the length of the shortest path from u to v which takes exactly k steps. The path is allowed to visit vertices multiple times (for example, the path $2 \rightarrow 5 \rightarrow 7 \rightarrow 5 \rightarrow 4$ is a valid path from 2 to 4 of length 4, even though it visits vertex 5 twice). What is the running time of your algorithm? You do not have to write the code for it.
2. Suppose you are given an array A[1..n] of sorted integers that has been circularly shifted k positions to the right. For example, [35, 42, 5, 15, 27, 29] is a sorted array that has been circularly shifted k = 2 positions, while [27, 29, 35, 42, 5, 15] has been shifted k = 4 positions. We can obviously find the largest element in A in O(n) time. Describe an O(log n) algorithm based on the divide-and-conquer strategy to find the largest element in A. (You need to show that your algorithm's time complexity is O(log n))
3. Let G be an arbitrary weighted, directed graph. Please describe an algorithm to detect if G existed negative cycles. And explain the time complexity of your method.
4. If we modify the RELAX portion of the Bellman-Ford algorithm so that it updates d[v] and π[v] if $d[v] \geq d[u] + w(u, v)$ (instead of doing so only if d[v] is strictly greater than $d[u] + w(u, v)$), does the resulting algorithm still produce correct shortest-path weights and a correct shortest-path tree? Justify your answer.
5. As we known CLIQUE problem is a NP-Complete problem, please proof that a vertex cover problem is also a NP-Complete problem.
6. Please use Huffman code to encode a text with the letters with the number of occurrences: 'b'(12), 'c'(3), 'e'(57), 'i'(51), 'o'(33), 'p'(20). And how many bits will you use to present the message? In comparison with fixed-length code, how many percentages of storage will you save?
7. Show that quicksort's best-case and mergesort's worst-case running time is $\Omega(n\log n)$