Department of Computer Science and Engineering
National Sun Yat-sen University
Design and Analysis of Algorithms - Final Exam., Jan. 8, 2019

1. Explain each of the following terms. (20%)
   (a) NP-complete
   (b) branch and bound
   (c) lower bound of a problem
   (d) 2-D ranking
   (e) skew heap
2. In the FFT algorithm, we need to calculate ω^t often. It is known that
   $e^{i\theta} = \cos\theta + i\sin\theta$
   $\omega = e^{i(2\pi)/n}$.
   Please give the values of ω^n and $\omega^{n/2}$. (8%)
3. Design an algorithm for solving the *knapsack* problem (not 0/1 knapsack problem). And analyze the time complexity of your algorithm. (Hints: greedy method) (12%)
4. In the self-organizing sequential search heuristics, what are the *transpose* heuristics, *move-to-front* heuristics and *count* heuristics? (12%)
5. Use the *prune-and-search* approach to design an algorithm for selecting the kth smallest element among n input elements. Your algorithm should be with $O(n)$ time. (12%)
6. Given n matrices $A_1, A_2, \ldots, A_n$ with size $p_0 \times p_1, p_1 \times p_2, p_2 \times p_3, \ldots, p_{n-1} \times p_n$, respectively, design a *dynamic programming* method to determine the multiplication order of $A_1 \times A_2 \times \ldots \times A_n$ such that the number of scalar multiplications is minimized. Note that for computing $A_i \times A_{i+1}$, it requires $p_{i-1}p_ip_{i+1}$ scalar multiplications. (12%)
7. An approximate algorithm for solving the *node cover* problem of a graph $G = (V, E)$ is given as follows. Let N denote the solution (node cover). Initially, $F = E$. Arbitrarily select an edge $e = (u,v) \in F$, next add nodes u and v into N. Then remove all edges incident to u or v from F. Repeat the above procedure until F becomes empty. Suppose that P is the optimal solution (node cover). Show that $|N| \leq 2|P|$. (12%)
8. Prove that the *Hamiltonian decision* problem polynomially reduces to the *traveling salesperson decision* problem. (12%)