Department of Computer Science and Engineering
National Sun Yat-sen University
Second Semester of 2023 PhD Qualifying Exam

Subject：Algorithms

1. Explain each of the following terms: *NP*, *NP-complete*, *NP-hard*. (15%)
2. Design an algorithm to find both the *minimum* and the *maximum* of n elements with at most $\lceil 3n/2 \rceil$ comparisons. (15%)
3. Explain the *breadth-first* search, *depth-first* search, and *best-first* search in the tree searching strategies. What data structures are used for them? (15%)
4. (a) What is the difference between the *knapsack* problem and the *0/1 knapsack* problem. (5%)

   (b) Present an algorithm for solving the *knapsack* problem. Analyze the time complexity of your algorithm. The complexity should be with $O(n^2)$, where n denotes the number of objects. You will get no point if your complexity exceeds $O(n^2)$. (10%)
5. Based on the prune-and-search approach, please design an algorithm for selecting the kth smallest element among n data elements. Your algorithm should be with $O(n)$ time. (15%)
6. Prove that the *sum of subset decision* problem polynomially reduces to the *partition* problem. (15%)
7. Given a sequence of integers $a_1, a_2, \ldots, a_n$ (positive, zero, or negative), the *maximum segment* problem aims to find a segment $a_i, a_{i+1}, \ldots, a_j$, $1 \le i \le j \le n$, such that $a_i + a_{i+1} + \ldots + a_j$ has the maximum value. In the given sequence, at least one integer is positive. Please design an algorithm for solving the problem in linear time. Analyze the time complexity. Note that you will get no point if the time complexity of your algorithm is not linear. (10%)