國立中山大學資訊工程學系
106 學年度第 1 學期博士班資格考試

科目：演算法

1. In the *self-organizing sequential search* heuristics, what are the *transpose* heuristics, *move-to-front* heuristics and *count* heuristics? (10%)
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. In the s*olution tree searching* strategy, explain each of the following and give the data structure used by each. (15%)
   (a) *depth-first search*
   (b) *breadth-first search*
   (c) *best-first search*
4. Given matrices A and B with sizes $p\times q$ and $q\times r$, respectively, the computation $C=A\times B$ requires $p\times q\times r$ scalar multiplications, and the size of C is $p\times r$. Given n matrices $A_1, A_2, \ldots, A_n$ with sizes $p_0\times p_1$, $p_1\times p_2$, $p_2\times p_3$, $\cdots$, $p_{n-1}\times p_n$, respectively, the *matrix-chain multiplication* problem is to determine the multiplication order such that the number of scalar multiplications is minimized. Let $m(i, j)$ denote the minimum number of scalar multiplications for computing $A_i \times A_{i+1} \times \ldots \times A_j$. Please present the *dynamic programming* formula for calculating $m(i, j)$. (15%)
5. In the *bin packing* problem, we are given n objects with sizes $a_1, a_2, \cdots, a_n$, $0<a_i\leq 1$, and a unbounded number of bins, each with one unit capacity. The *first-fit* algorithm is a simple approximation algorithm for solving the problem. It puts an object into the ith bin as long as the ith bin is available and tries the $(i+1)$th bin if otherwise. Show that the number of bins resulting from the first-fit algorithm is no more than twice the number of bins needed in an optimal solution. (15%)
6. The *sorting* problem is to arrange the input sequences into the nonincreasing (or nondecreasing) order. Please rewrite the sorting problem as the decision version of sorting problem. (15%)
7. Prove that the *Hamiltonian cycle* decision problem polynomially reduces to the *traveling salesperson* decision problem. (15%)