國立中山大學100學年度學士班轉學生招生考試試題

科目：離散數學【資訊工程學系三年級】

Note: you should explain how you obtain the answers in detail or get no credit!

1. In how many ways can 10 (identical) dimes be distributed among five children if
   (a) [5%] there are no restrictions?
   (b) [5%] each child gets at least one dime?
   (c) [5%] the oldest child gets at least two dimes?

2. [10%] Prove that there are infinitely many primes.

3. [10%] Let $\Sigma = \{v, w, x, y, z\}$ and $A = \bigcup_{n=1}^{6} \Sigma^n$. How many strings in A have xy as a proper prefix?

4. [10%] Let $A = \{1, 2, 3, 6, 9, 18\}$, and define relation R on A by xRy if $x|y$. Draw the Hasse diagram for the poset (A, R).

5. [10%] Let $A = \{1, 2, 3, 4, 5\}\times\{1, 2, 3, 4, 5\}$, and define R on A by $(x_1, y_1)R(x_2, y_2)$ if $(x_1+y_1) = (x_2+y_2)$, where R is an equivalence relation on A. Please determine the equivalence class $[(2, 4)]$.

6. Solve the following recurrence relations.
   (a) [5%] $a_n = 5a_{n-1} + 6a_{n-2}$, $n \geq 2$, $a_0 = 1$, $a_1 = 3$.
   (b) [5%] $a_n - 6a_{n-1} + 9a_{n-2} = 0$, $n \geq 2$, $a_0 = 5$, $a_1 = 12$.
   (c) [5%] $a_{n+2} + 3a_{n+1} + 2a_n = 3^n$, $n \geq 0$, $a_0 = 0$, $a_1 = 1$.

7. [10%] Find an integer m such that $0 < m < 23\cdot 29\cdot 31$ and
$$\begin{cases} m \equiv 1 \pmod{23} \\ m \equiv 0 \pmod{29} \\ m \equiv 2 \pmod{31} \end{cases}$$
by the Chinese Remainder Theorem.

8. [10%] Let G be a cyclic group. Prove that G is isomorphic to $(\mathbf{Z}, +)$ if $|G|$ is infinite.

9. [10%] Prove that every subgroup of a cyclic group is cyclic.

國立中山大學100學年度學士班轉學生招生考試試題

科目：資料結構【資訊工程學系三年級】

1. (25%) Let $A = [a_1, a_2, \ldots, a_n]$ be an array of n distinct positive integers. A is called a *heap*, if

$$a_i \geq \max\{a_{2i}, a_{2i+1}\}, \text{ for } i = 1, 2, \ldots, \lfloor n/2 \rfloor.$$

Note that if n is even, we define $a_{n+1} = 0$ in the above definition.

   (a) Show that a_1 is the maximum element in a heap A.

   (b) Can you design a *linear time* algorithm to print out the elements of a heap A in increasing or decreasing order? Justify your answer.

2. (25%) An undirected graph $G = (V, E)$ consists of a set of vertices V and a set of edges E. Assume that the graph is simple, that is, each edge $e \in E$ connects two distinct vertices in V and no two edges connect the same pair of vertices. Also assume that the graph is connected, that is, there is a path between every pair of vertices. It is known that a simple and connected graph will have at least $|V| - 1$ edges and at most $\binom{|V|}{2}$ edges. A graph G is *dense* if the number of edges is closed to $\binom{|V|}{2}$. A graph G is *sparse* if the number of edges is closed to $|V| - 1$. Design a data structure for dense graphs and a data structure for sparse graphs so that the memory required will be as small as possible.

3. (25%) Let S be a set of n elements. The *union-find* problem on the set S consists of a sequence of `union(x,y)` and `find(z)` instructions. Initially, each element in S is a set by itself. The instruction `union(x,y)` makes the set containing x and the set containing y into one set. The `find(z)` reports the name of the set in which z belongs. The name of a set can be any elements in that set, but it must be consistent. That is, any element in the same set should get the same name, and the name cannot be changed, except it is unioned to another set. Design data structures and algorithms for the problem so that both instructions can be executed efficiently for large $|S|$. Analize the time complexity of `union(x,y)` and `find(z)`.

4. (25%) Let $G = (V, E)$ be a connected graph. Let $w : E \rightarrow R^+$ be a nonegative weight function defined on the edges of G. Let H be a subgraph of G. The weight of H is defined to be the sum of the weights on the edges of H. A minimum spanning tree of G is a spanning tree with mimimum weight. The Prim's algorithm and the Kuskal's algorithm are the most commonly used algorithms to find a minimum spanning tree for G. Described the two algorithms and design a data structure for each of these two algorithms so that each of the algorithm can be run efficiently.