銘傳大學 98 學年度研究所碩士班招生考試

傳播管理研究所碩士班（丙組）、資訊管理學系碩士班

第二節

資料結構試題

（限用答案本作答）　☐可使用計算機 ☑不可使用計算機

1. Give a brief description of the following terms:
   (1) Recursion (5%)
   (2) Iterator (5%)
   (3) Priority queue (5%)
   (4) Hash table (5%)

2. A binary tree can be represented by an array with the following procedures:
   ROOT(): return 1
   LEFT-CHILD(i): return $2i$
   RIGHT-CHILD(i): return $2i + 1$
   PARENT(i): return $\lfloor i/2 \rfloor$
   (1) Show that $i \le 2^n - 1$ for a binary tree with n nodes. (10%)
   (2) Give an example of a binary tree with five nodes that attains the above upper bound on i. (10%)

3. The following procedure BUILD_MAX_HEAP can be used in a bottom-up manner to convert an array $A[1..n]$, where $n = length[A]$, into an n-element max-heap. Please answer the following questions.

```
BUILD_MAX_HEAP(A):
    heap_size[A] ← length[A]
    for i ← ⌊length[A]/2⌋ downto 1
        do MAX_HEAPIFY(A, i)
```

```
MAX_HEAPIFY(A, i):
    l ← 2i
    r ← 2i + 1
    if l ≤ heap_size[A] and A[l] > A[i] then
        largest ← l
    else largest ← i
    if r ≤ heap_size[A] and A[r] > A[largest] then
        largest ← r
    if largest ≠ i then
        exchange A[i] ↔ A[largest]
        MAX_HEAPIFY(A, largest)
```

本試題兩面印刷

銘傳大學 98 學年度研究所碩士班招生考試

傳播管理研究所碩士班（丙組）、資訊管理學系碩士班

第二節

資料結構試題

（限用答案本作答）　　☐可使用計算機 ☑不可使用計算機

(1) Suppose that the array $A = \langle 34, 5, 75, 9, 18, 2, 26, 67, 23, 10\rangle$. Please show the resultant max-heap A after performing the BUILD_MAX_HEAP(A) procedure. (15%)

(2) Write a pseudo-code, called HEAP_SORT, which can reorder an array A of n numbers in an ascending order based on the procedures BUILD_MAX_HEAP and MAX_HEAPIFY. (15%)

(3) What is the worst-case running time of your algorithm? (10%)

4. The Prim's algorithm MST-Prim(G, w, r) is a method for minimum spanning tree problems. Please illustrate the execution of the Prim's algorithm (starting from the root vertex a) step by step on the graph G shown in Figure 1. (20%)

```
MST-Prim(G, w, r):
    for each u ∈ V(G) do    // V is a set of all vertices in graph G.
        key[u] ← ∞          // key[u] is the minimum weight of any edge connecting
                            // u to a vertex in the tree A.
        π[u] ← NIL          // π[u] denotes the parent of u.
    key[r] ← 0
    Q ← V[G]                // Q is a min-priority queue.
    while Q ≠ ∅ do
        u ← Extract-Min(Q)
        for each v ∈ Adj[u] do    // Adj[u] denotes the set of all vertices adjacent to u.
            if v ∈ Q and w(u, v) < key[v] then    // w(u, v) is the weight associated with
                                                  // the edge (u, v).
                π[v] ← u
                key[v] ← w(u, v)
```

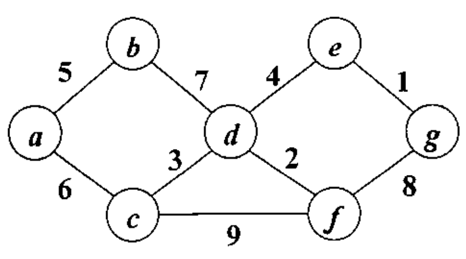

Figure 1. Weighted graph G.

本試題兩面印刷

試題完