國立中央大學97學年度碩士班考試入學試題卷

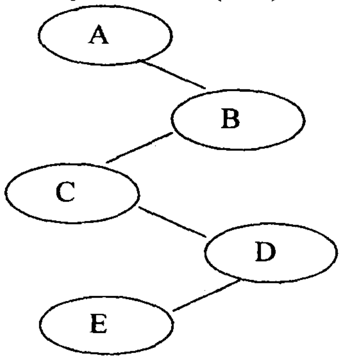
所別：資訊工程學系碩士班　科目：資料結構與演算法　共 2 頁　第 1 頁
軟體工程研究所碩士班　　　　　　　　　　　　　*請在試卷答案卷（卡）內作答

1. Below presents a formal description of Kruskal's algorithm. (10%)

```
Line 1    Start with an empty set T of edges.
Line 2    while (E is not empty && |T| != n-1)
          {
Line 3        Let (u,v) be a least-cost edge in E.
Line 4        E = E - {(u,v)}. // delete edge from E
Line 5        if ((u,v) does not create a cycle in T)
Line 6            Add edge (u,v) to T.
          }
Line 7    if (| T | == n-1) T is a min-cost spanning tree.
Line 8    else return no spanning tree.
```

Please use <u>two operations in the representation of sets</u> to present a method to check that the new edge, (u, v), does not form a cycle in T and add such an edge to T to produce a minimum spanning tree for Lines 5 and 6 above. <u>You will get zero score if you only use graph to answer whether the new edge, (u, v), forms a cycle in T.</u>

2. Given the binary tree below. (10%)



Please determine the order in which the nodes will be visited in the mixed order given by invoking function A:

```
void A( NODEPTR *root,
     void (*Visit)(char x))
{
 if (root){
    Visit(root->info);
    B(root->left, Visit);
    B(root->right, Visit);
    }
 }
```

```
void B( NODEPTR *root,
        void (*Visit)(char x))
{
if (root){
  A(root->left, Visit);
   Visit(root->info);
   A(root->right, Visit);
   }
}
```

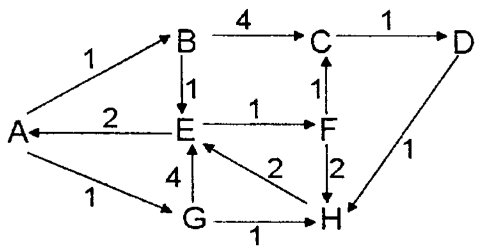
Where NODEPTR by declaring

```
typedef struct nodetype NODEPTR;
typedef struct nodetype {
     char info;
     nodetype *left;
     nodetype *right;
} nodetype;
```


3. Use the NODEPTR declaration in (2). Below is a recursive function in C that will interchange all left and right subtrees in a binary tree. Please write down the statements from (a) to (e) to complete the following function: (10%)

```
void swap(NODEPTR *root)
{
 NODEPTR   p;
   if (root) {
      ____(a)____;
      ____(b)____;
      ____(c)____;
      ____(d)____;
      ____(e)____;
   }
}
```

4. A directed graph G is given as follows (Notice that each edge has its weight.): (20%)



(a) Is G strongly connected? Why?

(b) Use Dijkstra's algorithm to find shortest path from A to D. Please show your process of getting the path.

(c) Use Depth-First-Search from Vertex A and follow the alphabetical order to explore the next vertex. List the set of back edges, the set of forward edges, and the set of cross edges.

5. Let A be an $n \times n$ matrix which consists of O's and X's such that, in any row of A, all the O's come before any X's in that row. Design an O(nlgn) algorithm to count the number of O's in A. (10%)

6. (a) Let T and P be two sequences $T[1..n]$ and $P[1..k]$ of characters, such that $k \le n$. Design an $O(n)$ time algorithm to determine whether P is a subsequence of T. Here, we say that P is a subsequence of T if there exist a sequence of indices $i_1 < i_2 \cdots < i_k$ such that for all j, $1 \le j \le k$, we have $T[i_j] = P[j]$. (10%)

(b) In the above problem (a) it is possible that there are many subsequences in T that match P. Now, assume that each character $T[i]$ in T has a cost $c[i]$ associated with it. Design an efficient algorithm to find the matching subsequence that maximizes the sum of costs. That is, find the sequence of indices $i_1 < i_2 \cdots < i_k$ such that for all j, $1 \le j \le k$, we have $T[i_j] = P[j]$ and $c[i_1] + c[i_2] + \ldots + c[i_k]$ is maximized. What is the time complexity of your algorithm? (10%)

7. The following statements are about minimum spanning trees (MST). Each statement may or may not be correct. In each case, either give an argument to prove it (if it is correct) or give a counterexample (if it isn't correct). Always assume that the graph $G=(V, E)$ is undirected and connected. Do not assume that edge weights are distinct unless this is specifically stated. <u>You will get zero score if you don't give any proving arguments or counterexamples.</u>

(a) If graph G has more than $|V|-1$ edges, and there is a unique heaviest edge, then this edge cannot be part of a MST. (5%)

(b) If graph G has a cycle with a unique heaviest edge e (that is, the weight of e is heaviest among those edges in the cycle and no other edge in the cycle has the same weight), then e cannot be part of any MST. (5%)

(c) If the lightest edge in a graph is unique, then it must be part of every MST. (5%)

(d) If graph G has a cycle with a unique lightest edge e, then e must be part of some MST. (5%)

注意：背面有試題