1. (15 points) Given a number sequence $S[1..n]$, the MaxSum procedure outputs $\max\{0, \max_{1\le i\le j\le n} \sum_{k=i}^{j} S[k]\}$. Complete the following pseudocode.

```
MaxSum (S, n)
begin
    x ← 0; y ← 0
    for i ← 1 to n
        do if S[i] + y > x then y ← [ 1A ]; x ← [ 1B ]
           elseif S[i] + y > 0 then y ← [ 1C ]
           else y ← 0
    print x
end
```

2. (15 points) Complete the following pseudocode so that it prints "123 123 132 123 123 213 213 231 213 123 321 321 312 321 123" in order when it is invoked by $X(S, 1, 3)$, where $S[i]=i$ for $1 \le i \le 3$.

```
X(S,k,n)
begin
  if k=n then print S[1..n]
  else
      for i ← k to n
        do temp←S[k]
           S[k]←S[i]
           S[i]←temp
           X(S,[ 2A ],[ 2B ])
           temp←S[k]
           S[k]←S[i]
           S[i]←temp
           X(S, n, n)
end
```

3. (20 points) **The binary-search-tree property:** Let x be a node in a binary search tree. If y is a node in the left subtree of x, then key[y] ≤ key[x]. If y is node in the right subtree of x, then key[x] ≤ key[y]. Suppose that we have integer numbers between 1 and 10000 in a binary search tree and want to search for the number 2006. The following sequence is the sequence of nodes examined. Give all the feasible ranges of the variable k in the sequence.

1000, 5566, 5203, k, 1314, 1510, 2381, 2006

4. (20 points) Give tight asymptotic bounds for the following recurrences.

a. $T(n) = 2T(\frac{n}{2}) + n$

b. $T(n) = T(\frac{n}{4}) + T(\frac{3n}{4}) + n$

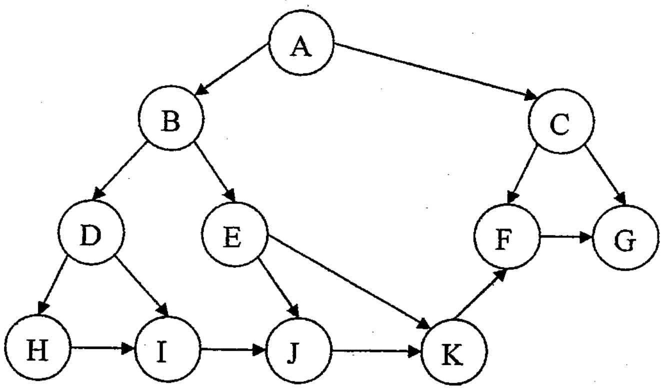
c. $T(n) = T(\frac{n}{7}) + T(\frac{3n}{4}) + n$

d. $T(n) = 7T(\frac{n}{4}) + n^2$

接背面

國立臺灣大學95學年度碩士班招生考試試題

5. (15 points) Add one edge to the following directed graph to make it strongly connected.



6. (15 points) Complete the following Depth-First Search pseudocode, which is used to classify the edges of the input graph G=(V, E):

```
DFS(G)
for each vertex  u ∈ V[G]
      do color[u] ← WHITE
            π[u] ← NIL
time ← 0
for each vertex  u ∈ V[G]
      do if color[u]=WHITE
            then DFS-VISIT(u)

DFS-VISIT(u)
color[u] ← GRAY
time ← time + 1
d[u] ← time
for each  v ∈ Adj[u]
      do if color[v]=WHITE
            then π[v] ← u
                  DFS-VISIT(v)
                  (u,v) is a tree edge
         elseif color[v]=GRAY
               then (u,v) is a [6A] edge
         elseif color[v]=BLACK
               then if [6B]
                        (u,v) is a cross edge
                     else (u,v) is a [6C] edge
color[u] ← BLACK
time ← time+1
f[u] ← time
```