复旦大学软件学院
2009～2010 学年第一学期期末考试试卷
■A 卷　　□B 卷

课程名称：数据结构与算法设计 II　　课程代码：SOFT130004.01
开课院系：软件学院　　考试形式：闭卷
姓　名：　　学　号：　　专　业：

题 号	1	2	3	4	5	6	7	8	总 分
得 分									

（装订线内不要答题）

Questions (100 points)

1. A sequence of stack operations is performed on a stack whose size never exceeds k. After every k operations, a copy of the entire stack is made for backup purposes. Show that the cost of n stack operations, including copying the stack, is $O(n)$ by assigning suitable amortized costs to the various stack operations. (5 points)

2. Describe in detail the topological sort algorithm, and what the running time it is when using it to sort the vertices of a DAG. Can the topological sort be used to solve the single-source shortest path problem for a DAG with negative edge path costs? Why? (10 points)

3. Give an algorithm to find the minimum number of edges that need to be removed from an undirected graph so that the resulting graph is acyclic. Show that this problem is NP-complete for directed graphs? (10 points)

4. What are the two key ingredients that an optimization problem must have in order for dynamic programming to be applicable? Beneath every greedy algorithm, there is almost always a more cumbersome dynamic-programming solution. If we can demonstrate what properties the problem has, then we are well on the way to developing a greedy algorithm for it? (10 points)

5. Write pseudocode for Prim's algorithm (Minimum spanning tree) and for the procedure MAKE-SET(x), UNION(x, y) and FIND-SET(x), which performs the corresponding manipulations on disjoint sets. (10 points)

6. Consider the following recursive algorithm for finding the shortest weighted path in an acyclic graph, from s to t. (15 points)

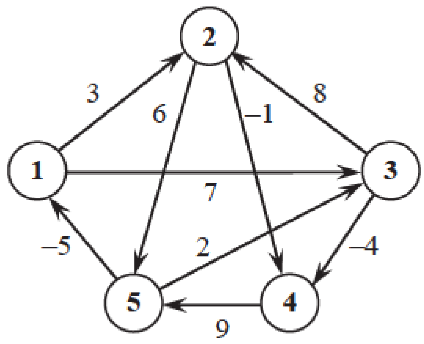
a. Why does this algorithm not work for general graphs?

b. Prove that this algorithm terminates for acyclic graphs.

c. What is the worst-case running time of the algorithm?

```
Distance shortest(s, t) {
   Distance d_t, tmp;
   If (s == t)
      return 0;
   d_t = ∞;
   for each Vertex v adjacent to s {
      tmp = shortest(v, t);
      if (c_s,v + tmp < d_t)
         d_t = c_s,v + tmp;
   }
   return d_t;
}
```

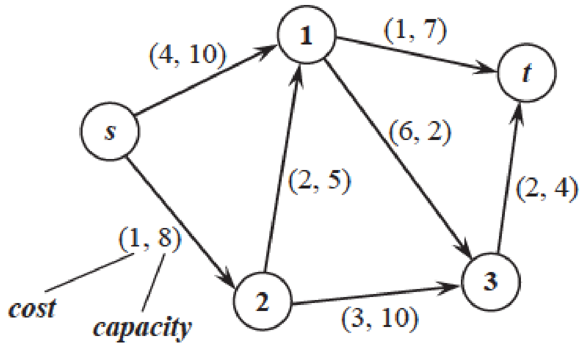
7. Find all-pairs shortest paths for the following graph by Floyd-Warshall algorithm (15 points)



8. Fill the blanks in the following pseudocode of Knuth-Morris-Pratt matching algorithm. (10 points)

```
KMP-MATCHER(T, P)
n ← length[T]
m ← length[P]
π ← COMPUTE-PREFIX-FUNCTION(P)
q ← 0    // Number of characters matched.
for i ← 1 to n
    do while ____(a)____ and P[q + 1] ≠ T[i]
            do ____(b)____
        if P[q + 1] = T[i]
            then ____(c)____
        if ____(d)____
            then print "Pattern occurs with shift" i − m
                ____(e)____
```

9. Find the minimum cost flow and the corresponding total cost which can ship 10 units from the source s to the sink t in the following network. Hint: if y is the cost of sending one unit of flow across the edge (u, v), then the cost of sending one unit of flow across the edge (v, u) is $-y$. (15 points)



Example Answers of Data Structures and Algorithms

(Software School, Fundan University, winter, 2009 - 2010)

Questions (100 points)

1. Amortized cost

PUSH 2
POP 0
BUCKUP 0

2. Topological sort

Step 1: Call depth first search algorithm.
Step 2: Sort by finishing time.
Run time: $O(n\lg n)$
Yes, it can be used to solve the single-source shortest path problem for a DAG with negative edge path costs because DAG is a directed acyclic graph.

3. Find the minimum spanning tree for every connected, undirected graph

Count the number of edges that appear in the minimum spanning trees, denoted by x. The minimum number of edges that need to be removed from an undirected graph that the resulting graph is acyclic is $E-x$.

4. Two key ingredients

Optimal substructure: Dynamic programming builds an optimal solution to the problem from optimal solutions to subproblems. The solutions to the subproblems used within the optimal solution to the problem must themselves be optimal by using a "cut-and-paste" technique. Subproblems should be independent.
Overlapping subproblems: Recursive algorithm revisits the same problem over and over again. In contrast, a problem for which a divide-and-conquer approach is suitable usually generates brand-new problems at each step of the recursion.

For greedy algorithm, we should prove that at any stage of the recursion, one of the optimal choices is the greedy choice. Thus, it is always safe to make the greedy choice.
Greedy-choice property: A globally optimal solution can be arrived at by making a locally optimal choice (a greedy choice at each step yields a globally optimal solution).

5. Write pseudocode for Prim's algorithm

```
MST-PRIM(G, w, r)
for each u ∈ V[G]
    do key[u] ← ∞
       π[u] ← NIL
key[r] ← Ø
Q ← V[G]
while Q ≠ 0
```

```
      do u ← EXTRACT-MIN(Q)
         for each v ∈ Adj[u]
            do if v ∈ Q and w(u, v) < key(v)
                  then π[v] ← u
                       key[v] ← w(u, v)
```

MAKE-SET(x)

```
p[x] ← x
rank[x] ← 0
```

FIND-SET(x)

```
if x ≠ p[x]
   then x ← FIND-SET(p[x])
return p[x]
```

UNION(x, y)

```
LINK(FIND-SET(x), FIND-SET(y))
```

LINK(x, y)

```
if rank[x] > rank[y]
   then p[y] ← x
   else p[x] ← y
        if rank[x] = rank[y]
           then rank[y] ← rank[y] + 1
```

6. Answer questions

a. This algorithm will not work for cyclic graphs

b. The algorithm is equal to DFS.

c. $O(V + E)$

7. Find all-pairs shortest paths

$$D = \begin{bmatrix} 0 & 3 & 7 & 2 & 9 \\ 1 & 0 & 8 & -1 & 6 \\ 0 & 3 & 0 & -4 & 5 \\ 4 & 7 & 11 & 0 & 9 \\ -5 & -2 & 2 & -3 & 0 \end{bmatrix} \qquad \pi = \begin{bmatrix} \varnothing & 1 & 1 & 2 & 2 \\ 5 & \varnothing & 5 & 2 & 2 \\ 5 & 1 & \varnothing & 3 & 4 \\ 5 & 1 & 5 & \varnothing & 4 \\ 5 & 1 & 5 & 2 & \varnothing \end{bmatrix}$$

8. Fill the blanks

(a) $q > 0$

(b) $q \leftarrow \pi\,[q]$

(c) $q \leftarrow q + 1$

(d) $q = m$

(e) $q \leftarrow \pi\,[q]$

9. Minimum cost flow

Path: $s \to 2 \to 1 \to t$, flow: 5, cost: 4, total: 20.

Path: $s \to 1 \to t$, flow: 2, cost: 5, total: 10.

Path: $s \to 2 \to 3 \to t$, flow: 3, cost: 6, total: 18.

Total cost: 48.