Data Structures and Object-oriented Programming
Instructor: Prof. Liwei Chan
Final Date:2021/06/07

Student ID: Student Name:

Part I (20%, 5% each)

1. (5%) Suppose we are debugging a quicksort implementation that is supposed to sort an array in ascending order. After the first partition step has been completed, the contents of the array are in the following order:

   3 9 1 14 17 24 22 30

   Which of the following statements is correct about the partition step?
   a. The pivot could have been either 14 or 17.
   b. The pivot could have been 14, but could not have been 17.
   c. The pivot could have been 17, but could not have been 14.
   d. Neither 14 nor 17 could have been the pivot.

2. (5%) Which of the below is the time required to merge two sorted lists of size m and n?
   a. O(m | n)
   b. O(m + n)
   c. O(m log n)
   d. O(n log m)
   e. none of the above

3. (5%) Which of the following statements about Kruskal algorithm and Prim's algorithm is correct?
   a. Kruskal algorithm's concentration is on edges.
   b. Prim's algorithm is better when there are many more vertices than edges.
   c. Kruskal algorithm can't use on negative-weighted-undirected graph.
   d. For Prim's algorithm, choose different vertex as start vertex may get different total weight of the minimum spanning tree.

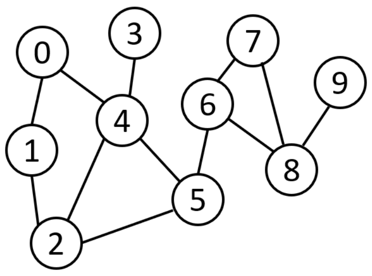
4. (5%) A symbol table is implemented using a Hash Table of table size 11 with hash function h(k) = k mod 11, where k is the key of a record.
   1. The records with keys: 13, 14, 2, 11, 18, 3, 36, 28, 21 are inserted in sequential order into the hash table using linear probing method for collision resolution. If keys 25 and 24 are searched, what is the total number of identifier comparisons?
      a. 11
      b. 13
      c. 15
      d. 17
      e. none of the above
   2. The records with keys: 13, 14, 2, 11, 18, 3, 36, 28, 21 are inserted in sequential order into the hash table using chaining method for collision resolution. What is the total number of identifier comparisons if each key value is searched once?

a. 11
b. 13
c. 15
d. 17
e. none of the above

Part II (80%, 10% each)

1. (10%) The content of an array is initially {320, 227, 139, 81, 992, 15, 308, 121, 78, 120}. Give the resulting array after **three main iterations of insertion sort**. You don't need to show details of computation.
   139, 227, 320, 81, 992, 15, 308, 121, 78, 120
   or 81, 139, 227, 320, 992, 15, 308, 121, 78, 120

2. (5%) According to the graph, please traverse the whole graph from Node 5 in the assigned method. If there are multiple routes on a vertex, choose the smaller one to print out first.

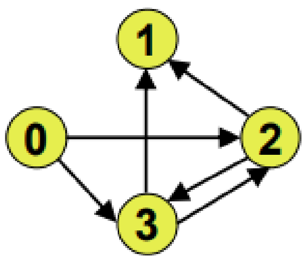


   a. (2.5%) Depth-First Search
      5 2 1 0 4 3 6 7 8 9

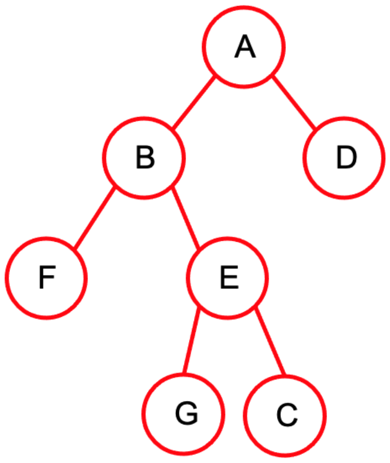
   b. (2.5%) Breadth-First Search
      5 2 4 6 1 0 3 7 8 9
      (5 2 4 6 <u>0 1</u> 3 7 8 9)也給對

3. (5%) For the following directed graph, fill in its adjacency matrix.

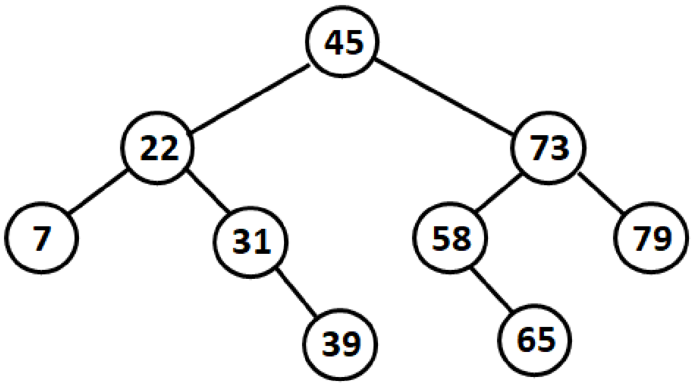
	0	1	2	3
0	0	0	1	1
1	0	0	0	0
2	0	1	0	1
3	0	1	1	0



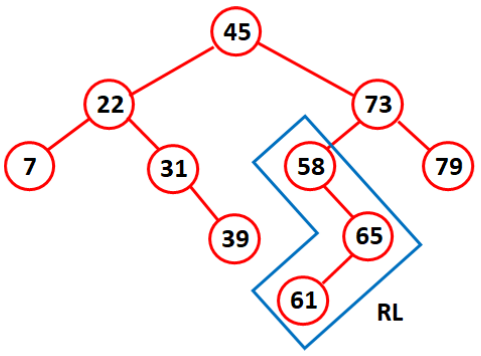
4. (10%) Please draw a binary tree according to its pre-order traversal and post-order traversal.
   Pre-order: A B F E G C D
   Post-order: F G C E B D A

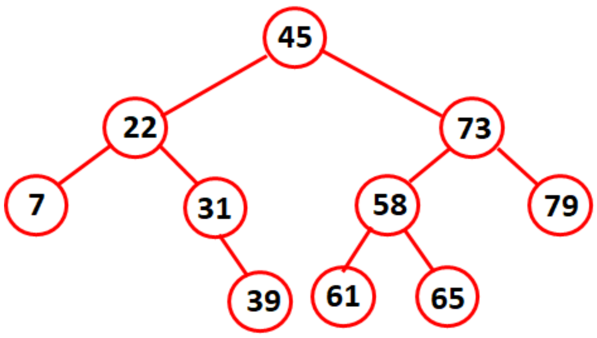


5. (10%)Given the following AVL tree, please answer the following questions and **draw the processing details as much as possible**.

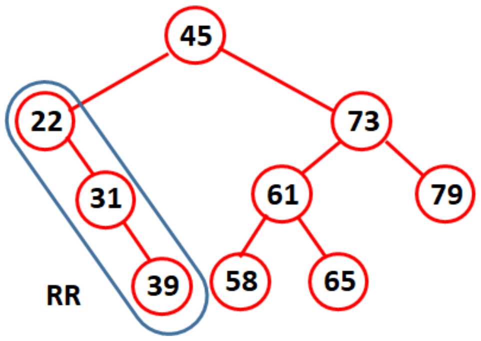


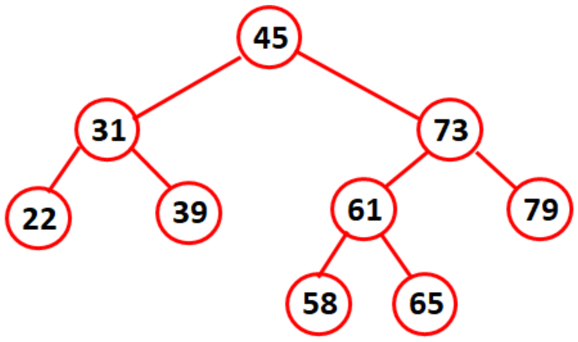
   a. (5%)Follow the given tree, **insert** “61” to the AVL tree. Draw the **processing details** and **results(**step-by-step figures**).**



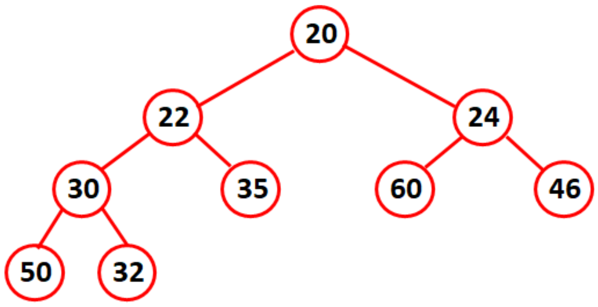


b. (5%)Follow the result of (a), **delete** “7” to the AVL tree.Draw the **processing details** and **results(**step-by-step figures)**.**

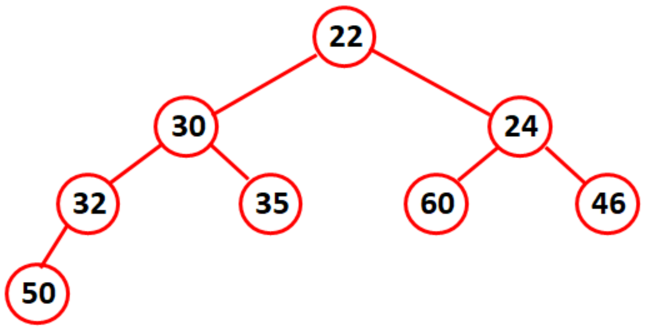




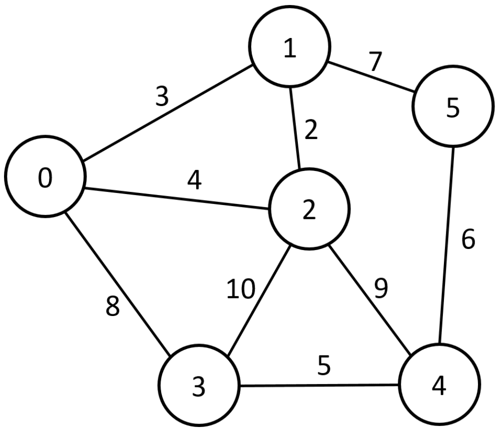
6. (10%)For the following question about **binary heap.**
   a. (5%)Build a **binary min-heap** by the insertion order : 35,32,46,50,22,60,24,30,20. **Draw the final result.**

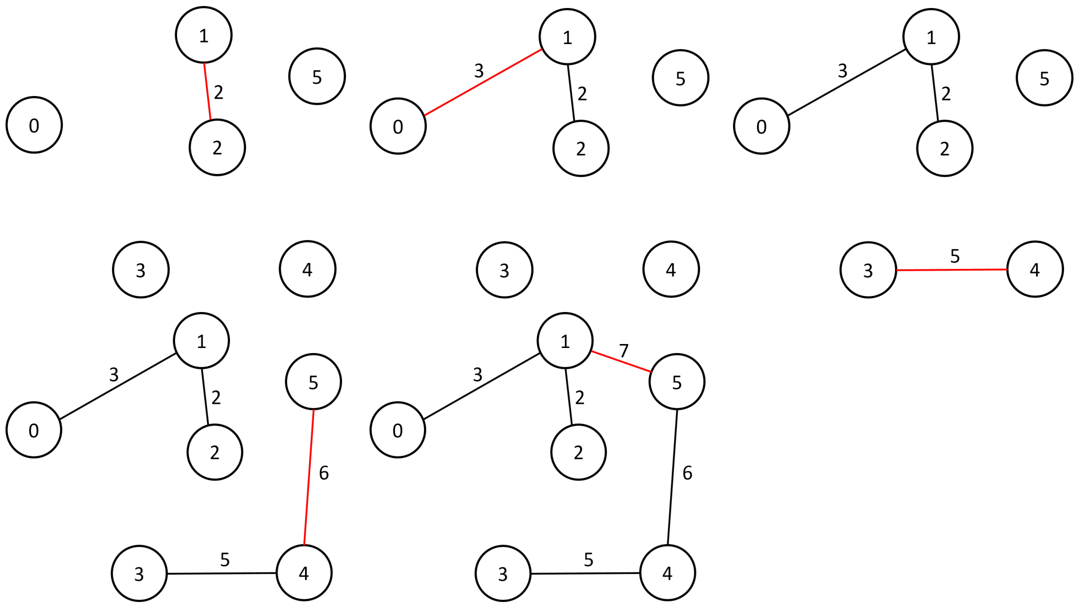


b. (5%)Follow the result of (a), delete the minimum node in the binary heap. **Draw the final result.**



7. (10%) Please refer to the following graph and solve the minimum spanning tree using **Kruskal's algorithm** step by step.





8. (10%) Please refer to the following pseudo code and finish the question below.

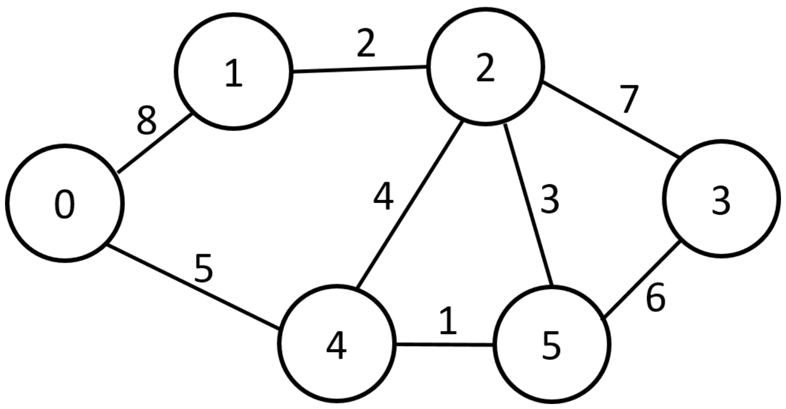
```
/*
G is a graph.
G.V is the set of vertices of G.
G.E is the set of edges of G.
w is the weight of the edge.
s is the source vertex.
d is distance.
p is parent vertex.
*/

UnknownFunction(G,w,s)
1    Initialize(G,s)
2    for i = 1 to |G.V|-1
3        for each edge (u,v) ϵ G.E
4            Relax(u,v,w)
5    for each edge(u,v) ϵ G.E
6        if v.d > u.d + w(u,v)
7            return FALSE
8    return TRUE


Initialize(G,s)
1    for each vertex v ϵ G.V
2        v.d = ∞
3        v.p = NIL
4    s.d = 0


Relax(u,v,w)
1    if v.d > u.d + w(u,v)
2        v.d = u.d + w(u,v)
3        v.p = u
```

a. (4%) What is the algorithm name of the UnknownFunction?
Bellman-Ford

b. (3%) What is the time complexity of line 2 to line 4 of the UnknownFunction (expressed in big-O)?
O(|G.V| * |G.E|)
(或者 O(|VE|)、O(G.V * G.E)、O(VE)也給對)

c. (3%) What is the time complexity of line 5 to line 7 of the UnknownFunction (expressed in big-O)?
O(|G.E|)
(或者 O(|E|)、O(G.E)、O(E)也給對)

9. (10%) Please consider the following graph, and answer the following question.



a. (5%) If you want to solve a single source shortest path problem, which algorithm will you choose among Bellman–Ford, Dijkstra's algorithm, Breadth-first search and briefly explain why.

Dijkstra's algorithm, since there is no negative weight in the graph, the time complexity of Dijkstra's algorithm is O(V^2) (O(V + ElogV) using binary heap). Which is faster than Bellman–Ford Algorithm(O(VE)). Breadth-first search could not solve the problem.

b. (5%) If we pick vertex 2 as the starting vertex of the Prim's algorithm, will it be different from choosing vertex 0? Please briefly explain why.

No, the minimum spanning tree is unique if all edges have distinct weights.