Algorithm Test

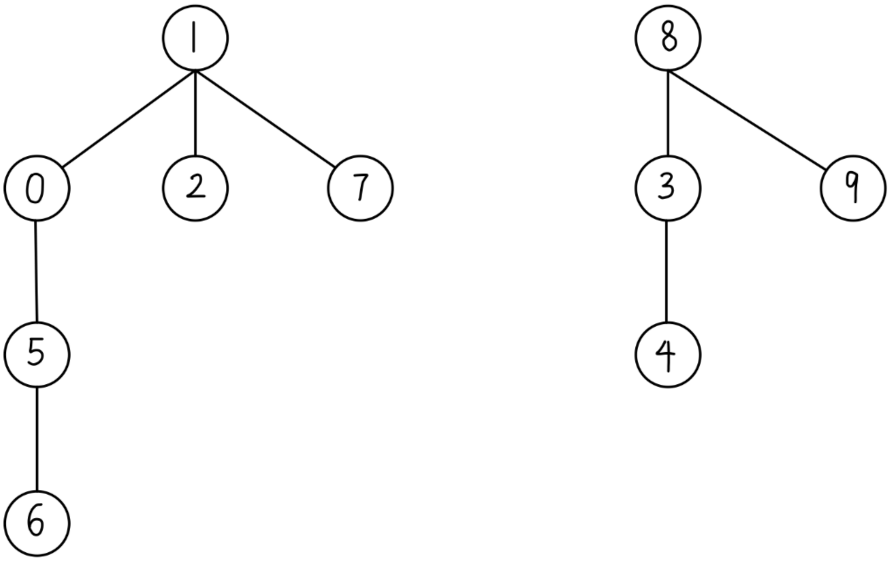
Problem 1. Analysis of algorithms. (15 points)

Give an array a[ ] that contains N integers, analyze the algorithm in the box below by using cost model and tilde notation to simplify operation count.

```
int count = 0 ;
for (int i = 0; i < N; i++)
      for (int j = i+1; j < N; j++)
            for (int k = j+1; k < N; k++)
                  if (a[ i ] + a[ j ] + a[ k ] == 0)
                        count++;
```

Problem 2. (10 points)

Construct a possible sequence of input pairs, such that the forest of trees below represented by the id [ ] array can be obtained by running quick-union. Note that, the detailed process of obtaining this forest should be given.



Problem 3. Sorting. (20 points)

Given an array: Q U I C K F I N D.

(1) Sort the characters in the array by using *bottom-up mergesort*.

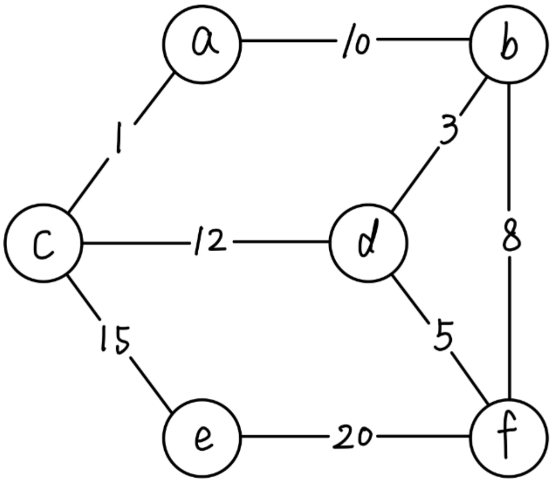
(2) In order to sort the characters in the array using *heapsort*, please give the process of building heap using *bottom-up* method.

Problem 4. Binary search trees. (10 points)

Give five orderings of the keys A X C S E R H that, when inserted into an initially empty BST, produce the *best-case* tree.

Problem 5. (15 points)

Give an edge-weighted graph G with 6 vertices and 8 edges. We use an algorithm learned in our course to get the MST of G by adding the edges to the MST in the following order: ac, bd, df, ab, ce.

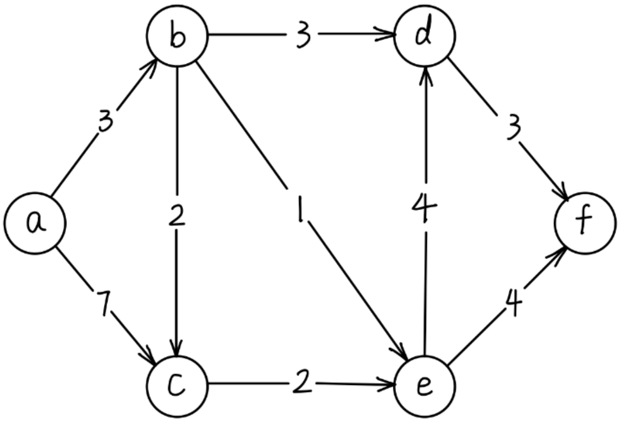


(1) Give the name of this algorithm and the data structures used to efficiently implement this algorithm.

(2) Give the process of using this algorithm to compute the MST of G.

Problem 6. (15 points)

Give an edge-weighted digraph as shown below, use Dijkstra's algorithm to compute the shortest path from vertex a to every other vertex.



(1) Draw the shortest path tree (SPT).

(2) Give the permutation of the vertices in the order that they are added to the SPT and give one possible permutation of the edges in the order that they are relaxed.

Problem 7. (15 points)

Give an array containing N characters that range from ‘a’ to ‘z’, design an algorithm to preprocess the input in liner time and then answers any query about how many of the N characters fall into a range [α..β] in constant time. Note that, α and β denote two characters in the range ‘a’ to ‘z’ and satisfy α≤β. Your answer will be graded on correctness, efficiency, clarity and conciseness.