装 订 线

任课教师：

装 订 线

学号：

装 订 线

姓名：

装 订 线

班级：

西 安 电 子 科 技 大 学

考试时间 120 分钟

试　　题

题号	一	二	三	四	五	六	七	总分
分数								

1. 考试形式：闭卷■　开卷□

2. 考试日期：　　年　　月　　日　（答题内容请写在装订线外）

Problem 1. Analysis of algorithms (15 points.)

Fill in the table to give the corresponding code framework of the given order of growth.

Order of growth	Code framework

N	`for (int i = 0; i < N; i++)`
	`{`
	`  …`
	`}`

NlgN	
N^2	

Problem 2. Union-Find. (10 points.)

Given the input pair sequence 0-6 6-1 4-5 6-4 2-4 6-3, draw the forest of trees represented by the id[] array after each input pair is processed until '0' and '5' are connected, by using *quick-union* and *weighted quick-union* respectively.

Problem 3. Sorting. (20 points. Two parts, 10 points for each part)

(1) Fill in the blanks to complete the implementation of merge operation in *Mergesort.*

```
void merge(Comparable[] a, Comparable[] aux, int lo, int mid, int hi)
{
    assert isSorted(a, lo, mid);
    assert isSorted(a, mid+1, hi);
    for (int k = lo; k <= hi; k++)
        aux[k] = a[k];
    int i = lo, j = mid+1;
    for (int k = lo; k <= hi; k++)
    {
        if      (i > mid)                  _____;
        else if (j > hi)                   _____;
        else if (less(aux[j], aux[i]))     _____;
        else                               _____;
    }
    assert isSorted(a, lo, hi);
}
```

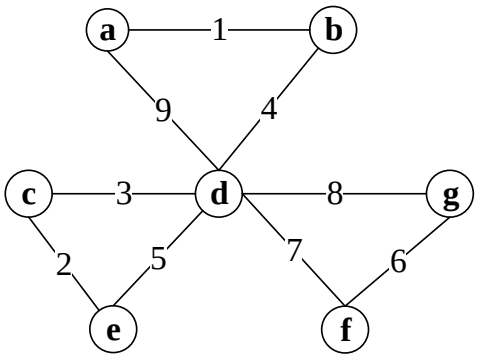
(2) Show the trace of the partition( ) in *Quicksort* working on array E D S Y F U E S.

Problem 4. Binary search trees (BST). (10 points)

Consider the keys `R E S P O N D` that, draw the best-case BST and the worst-case BST.

Problem 5. Minimum spanning trees (MST). (15 points. Two parts, 5 points for the first part and 10 points for the second part)

Given an edge-weighted graph G.

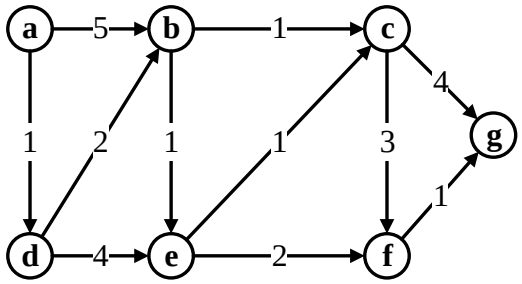


(1) Draw the MST of G.

(2) Give the process of using Kruskal's algorithm to compute the MST of G.

Problem 6. Shortest paths. (15 points. Two parts, 10 points for the first part, 5 points for the second part)

Give an edge-weighted digraph as shown below.



(1) Fill in the table below after running Dijkstra's algorithm to compute the shortest path from vertex a to every other vertex.

vertex	distTo[]	edgeTo[]
a	0	null
b		
c		
d		
e		
f		
g		

(2) Give all occurred values of distTo[e] and edgeTo[e] when running Dijkstra's algorithm to compute the shortest path from vertex a to every other vertex.

distTo[e]	edgeTo[e]

Problem 7. Algorithm design. (15 points.)

Given an array containing *N* characters that range from ‘a’ to ‘z’, design an algorithm to preprocess the input in linear time and then answer any query about how many of the *N* characters are equal to or greater than β in constant time. Note that, β denotes a character in the range ‘a’ to ‘z’. Your answer will be graded on correctness, efficiency, clarity, and conciseness.

附：单词释义表

题目	单词释义
1	order of growth 增长量级；
2	sequence 序列；respectively 分别地；
3	implementation 实现；
6	digraph 有向图；vertex 顶点；occurred 出现过的；
7	preprocess 预处理；query 查询；correctness, efficiency, clarity and conciseness 正确性、效率、清晰度和简洁性；