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	总分
分数							

1. 考试形式：闭卷■　开卷□

2. 考试日期：20　年　月　日(答题内容请写在装订线外)

Problem 1. Analysis of algorithms. (17 points)

(1) For each code fragment, give the best matching order of growth of the running time.

	int sum = 0; **for (int i = 1 i < N; i *= 2)** ** for (int j = 0; j < N; j++)** ** sum++;**	**A. lg *N*** **B. *N* + *R*** **C. *N* lg *N*** **D. *N*** **E. *N* (*N* + *R*)** **F. *RN*** **G. 2^N**
	int sum = 0; **for (int n = N; n > 0; n /= 2)** ** for (int i = 0; i < n; i++)** ** sum++;**	
	int x = 1, i, j; **for (i = 0; i < N; i++)** ** for (j = 1; j < R; j++)** ** x = x * j;**	

(2) Observe the following running times for a program with an input of size *N*.

N	time
10,00	0.1 seconds
2,000	0.3 seconds
4,000	2.5 seconds
8,000	20.0 seconds
16,000	159.8 seconds

Estimate the running time of the program (in seconds) on an input of size *N* = 40,000.

_____seconds

Problem 2. Union-Find. (18 points)

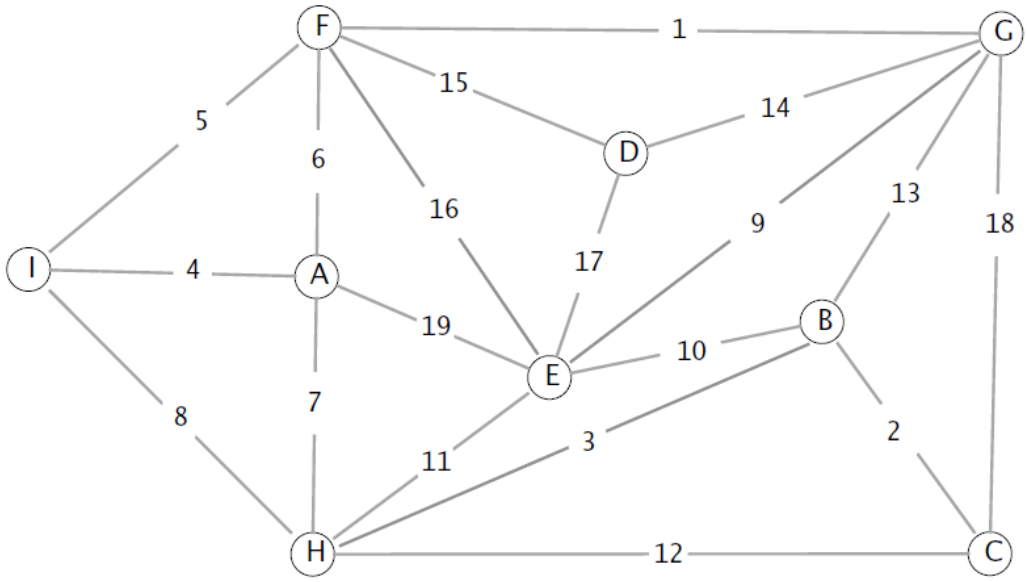
Show the contents of the id[] array and draw the forest of trees represented by the id[] array for each input pair by using *weighted quick-union* for the sequence 2-1 0-3 1-4 0-6 5-4 2-6. Note that, when processing an input pair p-q, if p and q are in two trees of the same size, you should link root of the tree containing p to root of the tree containing q.

Problem 3. Binary search trees. (12 points)

Draw the BST that results when you insert the keys E A S Y Q U E S T I O N, in that order (associating the value i with the ith key) into an initially empty tree.

Problem 4. Minimum spanning trees. (16 points)

Consider the following edge-weighted graph with 9 vertices and 19 edges. Note that the edge weights are distinct integers between 1 and 19.



(1) Give the sequence of edges in the MST in the order that *Kruskal's algorithm* includes them (by specifying their edge weights).

(2) Give the sequence of edges in the MST in the order that *Prim's algorithm* includes them (by specifying their edge weights), starting from vertex *A*.

Problem 5. Sorting. (23 points)

(1) Consider the *first* call to key-indexed counting when running LSD string sort on the input array a[] of 20 strings. Recall that key-indexed counting is comprised of four loops. Give the contents of the integer array count[] after each of the first three loops (for indices between 'a' and 'g'); then, give the contents of the string array (for the indices 0-5 and 18-19) after the fourth loop.

i	a[i]
0	badge
1	freed
2	blurb
3	embed
4	basic
5	field
6	bluff
7	dwarf
8	fudge
9	climb
10	cycle
11	bleed
12	budge
13	crumb
14	cubic
15	cable
16	blend
17	cliff
18	bread
19	cache

c	count[] (*first*)	count[] (*second*)	count[] (*third*)
⋮	⋮	⋮	⋮
'a'			
'b'			
'c'			
'd'			
'e'			
'f'			
'g'			
⋮	⋮	⋮	⋮

i	a[i] (*fourth*)
0	
1	
2	
3	
4	
5	
6	*not required*
7	*not required*
8	*not required*
9	*not required*
10	*not required*
11	*not required*
12	*not required*
13	*not required*
14	*not required*
15	*not required*
16	*not required*
17	*not required*
18	
19	

(2) Suppose that you are sorting an array containing the following 7 equal keys (the subscript is not part of the key--its purpose is to uniquely identify each of the equal keys).

$$E_0 \quad E_1 \quad E_2 \quad E_3 \quad E_4 \quad E_5 \quad E_6$$

What is the result of running the standard version (from the textbook) of each of the following sorting algorithms?

Quicksort (no shuffle)	
Heapsort	

Problem 6. Algorithm design. (14 points)

Given two arrays a [] and b [], containing *M* and *N* distinct integers, respectively, (with $N \geqslant M$), design an algorithm to determine how many integers are in common between the two arrays. The running time of your algorithm should be proportional to *N*lg*M* in the worst case and use at most a *constant* amount of extra memory. Your answer will be graded on correctness, efficiency, clarity, and conciseness.

附：单词释义表

题目	单词释义
1	code fragment 代码片段； order of growth 增长数量级； estimate 估计；
2	weighted 加权的；sequence 序列；
3	associate 关联；
4	distinct 不同的；
5	recall 回忆；is comprised of 包含； subscript 下标；identify 确定；
6	respectively 分别地；determine 确定； be proportional to 正比于；extra 额外的； correctness, efficiency, clarity, and conciseness 正确性、高效性、清晰度以及简洁性。