請按順序作答

1. Please answer the following questions.

(1) (3%) Consider the failure function defined as below. Show the values of f(0), f(1), …, f(5) of string $p_0p_1p_2p_3p_4p_5$=ababaa.

$$f(j) = \begin{cases} \text{largest } 0 \le i < j \text{ such that } p_0...p_i = p_{j-i}...p_j \text{ and } p_{i+1} \neq p_{j+1} \\ -1 \qquad\qquad\quad \text{if there is no } i \ge 0 \text{ satisfying above} \end{cases}$$

(2) (3%) Show the result of inserting the following integers into an empty binary search tree.

10 15 2 9 18 4 7 6

(3) (3%) Please show the cost of the minimum-cost spanning tree of the following graph.
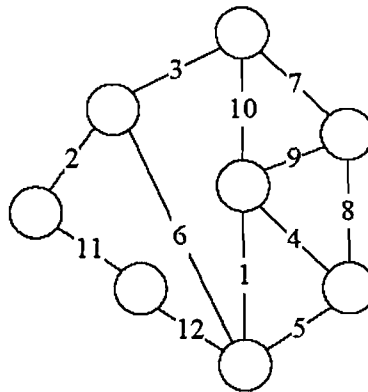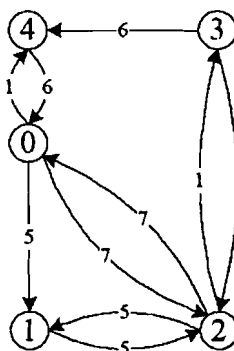
2. Please answer the following questions.

(1) (5%) Please show the result of the following program.

```c
#include <stdio.h>
void foo(int * a, int r, int n)
{
  int k=a[r];
  int j;
  for(j=2*r;k<=n;j*=2)
  {
    if (j<n)
      if (a[j]<a[j+1]) j++;
    if (k>=a[j]) break;
    a[j/2]=a[j];
  }
  a[j/2]=k;
}
int b[10]={4,7,1,5,15,12,9,6,3,8};
int n=10;
int main(void)
{
  int i;
  for(i=n/2;i>=1;i--)
    foo(b,i,n);
  for(i=0;i<10;i++)
    printf ("%d\t",b[i]);
}
```

(2) (5%) Consider a hash table ht[0..b-1] with b buckets, each bucket having one slot. Suppose a uniform hashing function with range [0, b-1] is used to hash n objects, say x_1, x_2,..., x_n, and chaining is used to handle overflow. Please calculate the average number of identifier comparisons needed to search a randomly chosen object x_j, $1 \leq j \leq n$.

(3) (5%) Let $A^k[i][j]$ be the length of the shortest path from i to j going through no intermediate vertex of index greater than k. Consider the following graph. Show the largest value of the non-infinity entries in matrix A^1.

國立交通大學 98 學年度碩士班考試入學試題

科目：資料結構與演算法(1001) *(9161) (4552)*   考試日期：98 年 3 月 15 日　第 1 節
系所班別：資訊系所跨組聯招、生資所乙組、資訊科技（IT）產專　　第 3 頁,共 10 頁
【不可使用計算機】＊作答前請先核對試題、答案卷（試卷）與准考證之所組別與考科是否相符！！

3. Answer the following 13 questions, 2% for each question. If the question asks for an answer, give me **ONLY** the answer. I don't need your reasons. If the question asks you to choose true statements from 4, the number of true statements is two. You get 2% only when you correctly identify the two. You don't lose any point if you fail to identify the correct answer. Number your answers as 3-(1), 3-(2) up to 3-(13). And put your asnwers consecutively in a block. **DO NOT** inter-mix with the other problem set.

(1) We abuse the "+" operator with the asymptotic notations. For example, we may say that the total time for an algorithm is $O(n) + \Theta(n)$. Which two of the following statements are true. (a) $O(n \log n) + \Theta(n^2) = \Theta(n^2)$. (b) $O(n^2) + \Theta(n^2) = \Theta(n^2)$. (c) $O(n \log n) + \Theta(n \log n) = O(n \log n)$. (d) $O(n^2) + O(n \log n) = \Theta(n^2)$.

(2) Given the recursion $T(1) = \Theta(1)$, $T(n) = a \cdot T(n/c) + n$, Solution to the recursion depends on the relationship between a and c. Which two of the following statements are true. (a) Solution is $\Theta(n \log n)$ if $a \le c$. (b) Solution is $\Theta(a^{\log_c n})$ if $a > c$. (c) Solution is $\Theta(n^{\log_c a})$ if $a > c$. (d) For any case, solution is greater that $O(n)$ but cannot be greater $O(n^2)$.

(3.) Any sorting algorithm needs at least $n \log n$ steps to sort n numbers if only comparison operators are allowed. Which two of the following statements are true. (a) Time for merge-sort can be modeled as the recursion $T(n) = 2T(n/2) + cn$ that has solution $\Theta(n \log n)$. And balance recursion achieves the best result. Thus the lower bound to sorting problem is $\Omega(n \log n)$. (b) It can be shown that at least $n \log n$ comparisons are needed in order to determine the permutation that the records are in increasing order. Thus $\Omega(n \log n)$ is the lower bound. (c) Quick-sort has the best performance among sorting algorithms if only comparisons are allowed. It can be shown that the average performance is $O(n \log n)$. No one can win the quick-sort so the best possible bound is $\Omega(n \log n)$. (d) Radix sort sorts n records in linear time because it employs operator that is not comparison.

(4.) Given the tree shown in Figure 3-1. Inorder traverse the tree. Which two of the following are correct (a) R is the second in the linear order. (b) S is the third in the linear order. (c) T is the fourth in the linear order. (d) V is the fifth in the linear order.

(5.) Suppose that we have n records, a_i, $i = 1, \ldots, n$, and these n recrods are stored in the nodes in a binary search tree. We call this kind of node the data node and each node (record) is associated with an access probability p_i. If a search in the binary search tree reaches an external node between a_i and a_{i+1}, we say that the search reaches a failure node. There are $n + 1$ failure nodes. Each failure node is associated with a probability q_i, $i = 0, \ldots, n$. A node (data node or failure node) contributes cost $p \cdot h$ to the total search cost where p is the associated probability and h is the depth of the node. The binary search tree stores these n recrods is an optimal binary search tree if the total cost $(\sum_i p_i \cdot h_i + \sum_j q_j \cdot h_j)$ is the least. Which two of the following statements are true. (a) Suppose there are 4 records with key values (10, 15, 20, 25), p_i are (3/16, 3/16, 1/16, 1/16), and q_i are (2/16, 3/16, 1/16, 1/16, 1/16), the optimal binary search tree is as shown in Figure 3-2. (b) Suppose there are 4 records with key values (10, 15, 20, 25),

p_i are (3/16, 3/16, 1/16, 1/16), and q_i are (2/16, 3/16, 1/16, 1/16, 1/16), the optimal binary search tree is as shown in Figure 3-3. (c) The optimal binary search is constructed by using the divide and conquer technique that can be done in $O(n \log n)$ time. (d) If there are n records and every node has the identical access probability, then the cost for the optimal binary search tree is $\Theta(n \log n)$.

(6) Forest representation for the Union-Find operations: given two sets shown in Figure 3-4, what is the result after Union of the two sets, if the weighting rule is applied. Given the set after Union, what is the result after the Find(i) operation if the collapsing rule is applied.

(7) The original leftist tree and the leftist tree after deleting minimum is shown in Figure 3-5. What are the key values should be in the nodes marked X and Y.

(8) Given the binary search tree shown in Figure 3-6, draw the resulted tree after the root is deleted. (A to J in the nodes are the "name" of the node, **NOT** the key values.)

(9) Given the AVL tree shown in Figure 3-7, draw the resulted tree after 35 is inserted.

(10) Given the 2-3 tree shown in Figure 3-8, draw the resulted tree after inserting 60 into the 2-3 tree.

(11) Given the Symmetric Min-Max Heaps (SMMH) shown in Figure 3-9, draw the resulted heap after the minimum is deleted.

(12) Consider the problem for optimal merging runs in external sort. We are given n runs, r_i, $i = 1, \ldots, n$. Run i has length l_i. To determine the optimal sequence of merging runs can be done by constructing a binary tree. Which two of the following statements are true? (a) This problem can be solved efficiently by using the dynamic programming. (b) If l_i=(1,2,3,9), the binary tree looks like Figure 3-10. (c) if l_i=(2,2,3,9), the binary tree looks like Figure 3-11. (d) This problem is the same as the Hoffman code for file compression.

(13) The following statements are about the heap sort. Which two of the statements are true. (a) Heap sort can be considered as an improvement of the selection sort. A heap structure is needed. Heap is a tree but can be stored in an array. (b) Two steps are involved, first is to construct the heap and the second is to sort. Both steps need $O(n \log n)$ time. Thus the total time is $O(n \, logn)$. (c) The sorting algorithm is in-place. (d) Standard inplementation of the heap sort algorithm is stable.
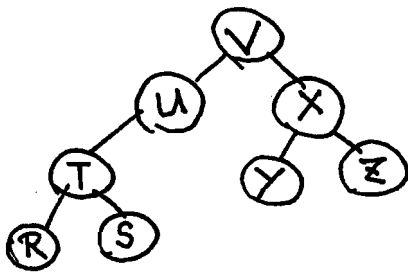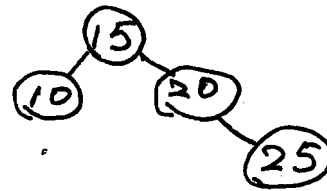
Figure 3-1

Figure 3-2

Figure 3-3

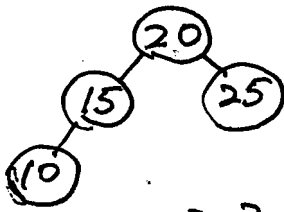Figure 3-4

Figure 3-5

Figure 3-6

Figure 3-7.

Figure 3-8
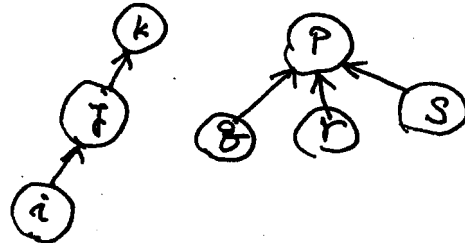
Figure. 3-9.

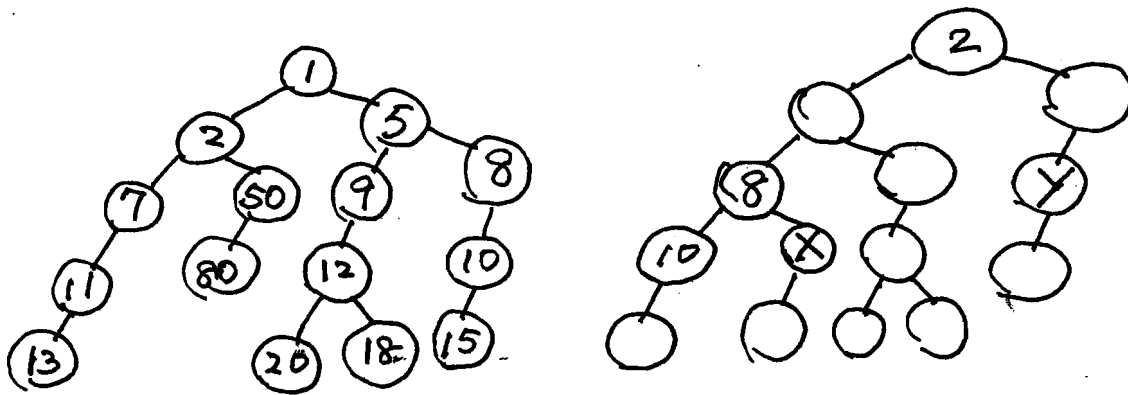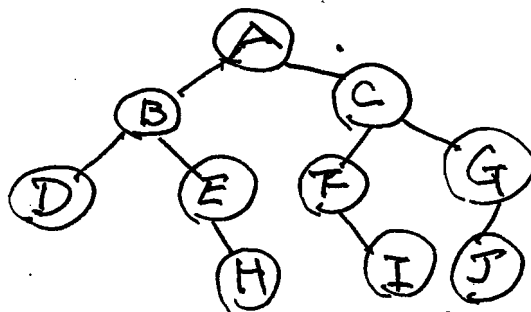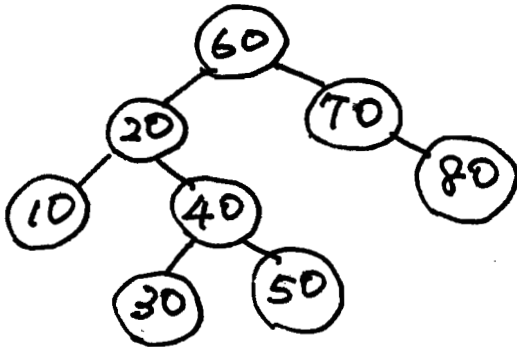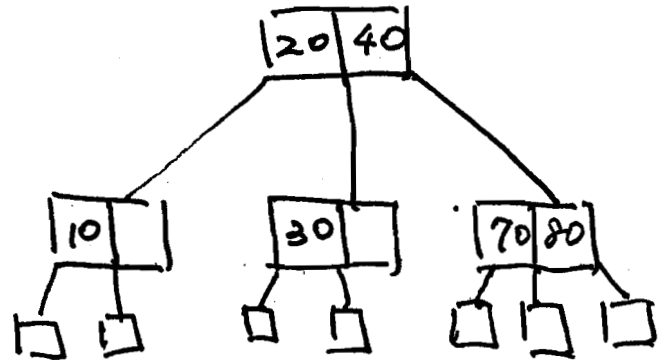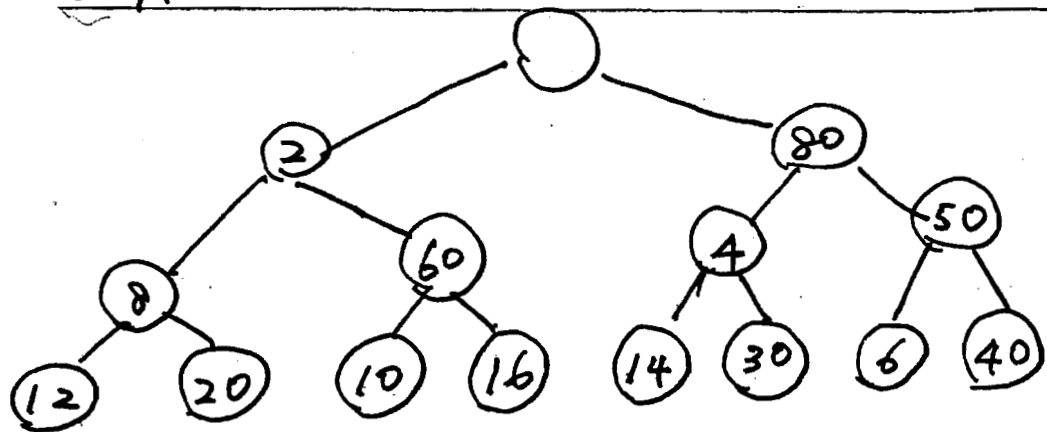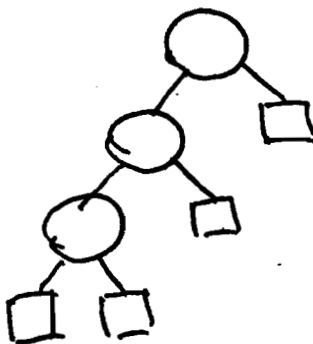Figure 3-10
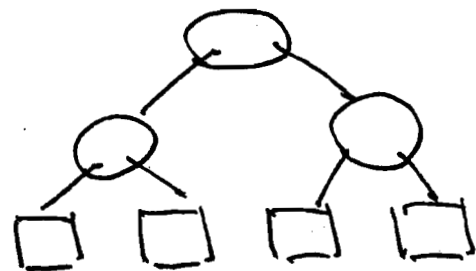
Figure 3-11

國立交通大學 98 學年度碩士班考試入學試題

科目：資料結構與演算法(1001) (9161) (4552)　　　　考試日期:98年3月15日 第 1 節
系所班別：資訊系所跨組聯招、生資所乙組、資訊科技（IT）產專　　　第 7 頁,共 10 頁
【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

4. (1)(5%) Let $P(n) = \sum_{i=0}^{d} a_i n^i$ be a degree-d polynomial in n, where $a_d > 0$,

and let k be a constant.

In the table below, indicate whether $P(n)$ is O, o, Ω, ω or Θ of n^k. Your answer

should be in the form of the table with "yes" or "no" written in each box.

		O	o	Ω	ω	Θ
(A) $P(n)$	$n^k, k > d$					
(B) $P(n)$	$n^k, k < d$					
(C) $P(n)$	$n^k, k = d$					

(2) (5%) Consider the following 15 functions. How many of them are polynomially bounded functions? (只要回答有幾個，不要寫理由。)

2^n, $n \log n$, $\sqrt{\log n}$, $\log^2 n$, $\log_2(n!)$, $\lceil (\log_2 n) \rceil!$, $n!$, $\lceil (\log \log n) \rceil!$,

$2^{\sqrt{2\log_2 n}}$, $(\sqrt{2})^{\log_2 n}$, $\sqrt{2^n}$, $n^{\log \log n}$, $(\log n)^{\log n}$, $4^{\log_2 n}$, $n^{\frac{1}{\log_2 n}}$,

(3) (5%) Give asymptotic tight bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for sufficiently small n. (只要寫出答案，不要寫理由或計算式。)

(A) $T(n) = 2T(\frac{n}{2}) + n \log n$

(B) $T(n) = 3T(\frac{n}{2}) + n \log n$

(C) $T(n) = 2T(\frac{n}{2}) + \frac{n}{\log n}$

(D) $T(n) = T(\lfloor \frac{n}{2} \rfloor) + 1$

(E) $T(n) = T(\frac{n}{2}) + \log n$

國立交通大學 98 學年度碩士班考試入學試題

科目：資料結構與演算法(1001) (9161)(4552)　　考試日期:98年3月15日　第 1 節
系所班別：資訊系所跨組聯招、生資所乙組、資訊科技（IT）產專　　第 8 頁,共 10 頁
【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

5.(10%) For the following questions (1) to (5), consider the 9 sorting algorithms listed below:

bubble sort,　bucket sort,　counting sort,　heap sort,　insert sort,

merge sort,　quick sort,　radix sort　and　selection sort.

（都不要寫理由，只要簡答。）

(1)　How many of them are comparison sort algorithms?

(2)　Among these 9 sorting algorithms, indicate those whose worst case time complexity are/is computed by the following recurrence.

$$T(n) = 2T(\frac{n}{2}) + O(n)$$

(3)　How many of them whose worst case time complexity are $\Theta(n \log n)$?

(4)　How many of them whose worst case time complexity are $\Theta(n^2)$?

(5)　How many of them whose worst case time complexity are linear time?

(6)　Consider the decision trees for comparison based sorting algorithms.

What is the smallest possible height of a decision tree for a comparison sort?

(7)　What is the average case time complexity of the quick sort.

(Give the best possible answer.)

(8)　Consider the comparison sort algorithms.

Is there any algorithm to sort n numbers in linear time in the worst case?

(9)　Consider the problem of finding median by comparison.

Is there any algorithm to find the median of n numbers in linear time in the worst case?

(10)　Is the sequence (23, 17, 14, 6, 13, 10, 1, 5, 7, 12) a max-heap?

If not, how many places that do not satisfy the property of the heap?

And indicate the numbers in those places.

6 Fill in the following 10 blanks. (25% in total).

Fractional and 0-1 knapsack

Given n items with values v_i and weights w_i, $1 \le i \le n$, and a knapsack with maximum weight W, where v_i, w_i, and W are positive integers.

(1) The fractional knapsack problem is to find the maximum value of $\sum_{i=1}^{n} v_i x_i$, subject to $\sum_{i=1}^{n} w_i x_i \le W$, where $0 \le x_i \le 1$.

This problem satisfies the greedy-choice property: Assume that

$$v_1/w_1 \ge v_2/w_2 \ge \ldots \ge v_n/w_n$$

there is an optimal solution with $x_1 = $ _____. (2%)

(2) The 0-1 knapsack problem is to find the maximum value of $\sum_{i=1}^{n} v_i x_i$, subject to $\sum_{i=1}^{n} w_i x_i \le W$, where $x_i = 0$ or 1.

This problem can be solved by dynamic programming. Define $c[i,w]$ to be the value of the solution for items 1, ..., i and maximum knapsack weight w where $i, w \ge 0$, Then, the recurrence of $c[i,w]$ is _____. (2%)

(3) The pseudocode _____ reduces the 0-1 knapsack decision problem to the 0-1 knapsack optimization problem. (2%)

Note: Fill in the blank with an algorithm that uses the following notations:

KNAPSACKOPT(v_i, w_i, W) denotes an integer-valued function that solves the 0-1 knapsack optimization problem with the instance v_i, w_i, and W.

KNAPSACKDEC(v_i, w_i, W, B) denotes a boolean-valued function that solves the 0-1 knapsack decision problem with the instance v_i, w_i, W, and B, where B is a positive integer lower bound.

(4) The pseudocode _____ reduces the 0-1 knapsack optimization problem to the 0-1 knapsack decision problem. (4%)

Note: Although you *needn't* prove it, your reductions for (3) and (4) must satisfy this property: The 0-1 knapsack decision problem belongs to P iff the 0-1 knapsack optimization problem can be solved in polynomial time.

MINCUT and MAXCUT

The MINCUT problem is to partition the vertices in a graph into two disjoint sets so that the number of edges between vertices in different sets is minimized.

(5) _____ Choose all and only those complexity classes that are known to contain the corresponding MINCUT decision problem. (2%)

① P　　② co-P　　③ NP　　④ co-NP　　⑤ NP-complete

國立交通大學 98 學年度碩士班考試入學試題

科目：資料結構與演算法(1001) (9161) (4552)　　　考試日期：98 年 3 月 15 日　第 1 節
系所班別：資訊系所跨組聯招、生資所乙組、資訊科技（IT）產專　　　第 /○ 頁,共 /○ 頁
【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

The MAXCUT problem is to partition the vertices in a graph into two disjoint sets so that the number of edges between vertices in different sets is maximized.

Consider the classic hill-climbing approximation algorithm for MAXCUT:

APPROX-MAXCUT($G = (V, E)$)
1　Start with the cut (S, T), where side $S = V$ and side $T = \varnothing$
2　Repeatedly move a vertex from one side to the other side, if this increases the number of edges in the cut
3　return the cut (S, T)

(6) The running time of this approximation algorithm is bounded from above by _____.　(The bound must be as tight as possible.)　(2%)

(7) The number of edges in the cut computed by this approximation algorithm is bounded from below by _____. (The bound must be as tight as possible.) (2%)

(8) The cut computed by this approximation algorithm depends on which hill it climbs up when there are several choices. The smallest possible number of edges in the cut computed by APPROX-MAXCUT($K_{2n,2n}$) is _____, where $K_{2n,2n}$ is a complete bipartite graph with $2n$ vertices on each side, $n \geq 1$. (4%)

Amortized analysis

A sequence of n operations is performed on a data structure.

Let c_i be the actual cost of the ith operation, $1 \leq i \leq n$

Define　$c_i = i$, if $i - 1$ is an exact power of 2
　　　　　 = 1, otherwise.

Let D_0 be the initial data structure, and D_i be the data structure that results after applying the ith operation to the data structure D_{i-1}.

Let $\Phi(D_i)$ be the potential associated with the data structure D_i

Define　$\Phi(D_0) = 0$
　　　　　$\Phi(D_i) = 2i - 2^{\lceil \log_2 i \rceil}$, $i \geq 1$.

Let $\hat{c}_i$ be the amortized cost of the ith operation with respect to Φ.

(9) The value of $\sum_{i=1}^{n}(\hat{c}_i - c_i)$ is _____.　(2%)

(10) The value of $\hat{c}_i$ is _____.　(Note: There are two cases.)　(3%)