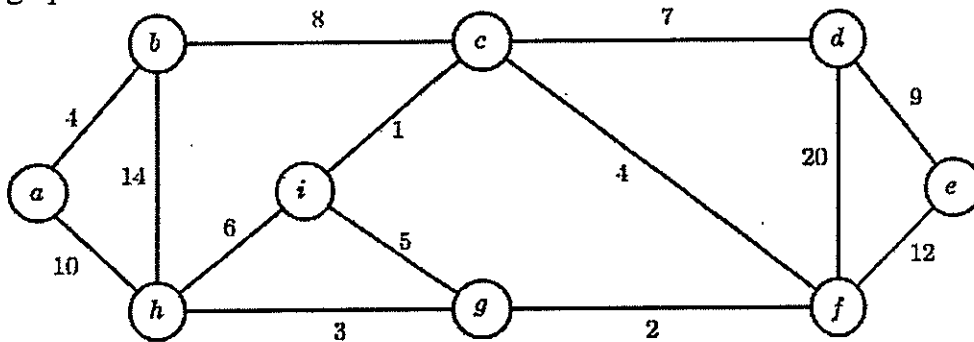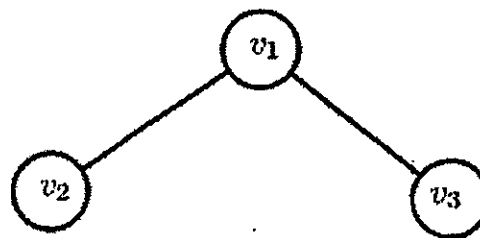※ 考生請注意：本試題不可使用計算機。　請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

一、資料結構：(50%)

1. (10%) Suppose that we have numbers between 1 and 1000 in a binary search tree, and we want to search for the number 363. Which of the following sequences could not be the sequence of nodes examined?
   a. 2, 252, 401, 398, 330, 344, 397, 363.
   b. 924, 220, 911, 244, 898, 258, 362, 363.
   c. 925, 202, 911, 240, 912, 245, 363.
   d. 2, 399, 387, 219, 266, 382, 381, 278, 363.
   e. 935, 278, 347, 621, 299, 392, 358, 363.

2. (10%) Compute the minimum weight spanning tree and it cost of the following graph.

3. (10%) Please write down the adjacency matrix of the following graph in the following table.

	v_1	v_2	v_3
v_1			
v_2			
v_3			

※ 考生請注意：本試題不可使用計算機。　請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

4. (10%) Please compute the total cost of executing n of the stack operations (PUSH, POP, and MULTIPOP).

Stack Operations: PUSH , POP , and MULTIPOP

PUSH (S, x) pushes object x onto stack S

POP (S) pops the top of stack S and returns the popped object. Calling POP on an empty stack generates an error.

MUTIPOP(S, k) removes the k top objectives of stack S, popping the entire stack if the stack contains fewer than k objects. The following pseudo code presents the procedure of the MUTIPOP operation

MUTIPOP (S, k)

1　　**while** not STACK -EMPTY (S) and $k > 0$
2　　　　POP (S)
3　　　　$k = k-1$

Both PUSH and POP take O(1) time. The cost of MUTIPOP (S, k) is $1 + \min(s, k)$ where s is the number of objects in stack S.

Suppose that there are n stack operations (PUSH, POP, and MULTIPOP) executed.

The stack is empty before those operations start and it is empty after those operations finished.

5. (10%) What is the maximum numbers of elements in a heap of height h?

※ 考生請注意：本試題不可使用計算機。 請於答案卷(卡)作答，於本試題紙上作答者，不予計分。

二、演算法：(50%)

6. (10%) If someone gives a polynomial algorithm for an NP-hard problem, what conclusions you can draw from this fact?

7. (10%) Given a sequence of n numbers, what is the lower bound of sorting algorithms using comparison and exchange operations?

8. (10%) The following graph is a maximum flow problem. Each edge is labeled with its capacity. What is the maximum flow from vertex s to vertex t.
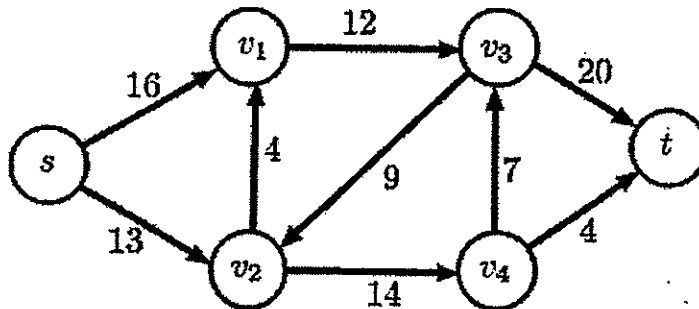
9. (10%) Use Master theorem to solve the recurrence $T(n) = 3T(2n/3) + O(1)$.

10. (10%) Given a graph $G = (V, E)$, a vertex cover $C \subseteq V$ is a vertex subsetet, such that if $(u, v) \in E$, then $u \in C$ or $v \in C$. The minimum vertex cover problem is to find a vertex cover C^* of G with the minimum cardinality. The following algorithm is a δ-approximation algorithm to find a vertex cover C satisfying $|C| \leq \delta \cdot |C^*|$. Please compute the minimum value of δ.

```
APPROX_VERTEX_COVER(G)
1 C ← 0
2 E' ← E{G)
3 while E' ≠ 0
4     do let (u, v) be an arbitrary edge of G
5         C ← C ∪ {u,v}
6         remove from E' every edge incident on either u or v.
7 return C
```