1. X is a two dimensional array. The address of X(4,2) is 1978 and X(2,3) is 1986. Assume a byte machine is used. Each element of X occupies two bytes.

   The address of X(3,8) is (a)2000 (b)2024 (c)2042 (d)2048 (5%)

   The number of rows of X is (a)6 (b)7 (c)8 (d)9. (5%)

   X is an array in (a)row-major (b)column-major (c)undecidable. (5%)

2. Suppose we are given the preorder sequence

   C A D E H B F G I

   and the inorder sequence

   D A H E B C F I G

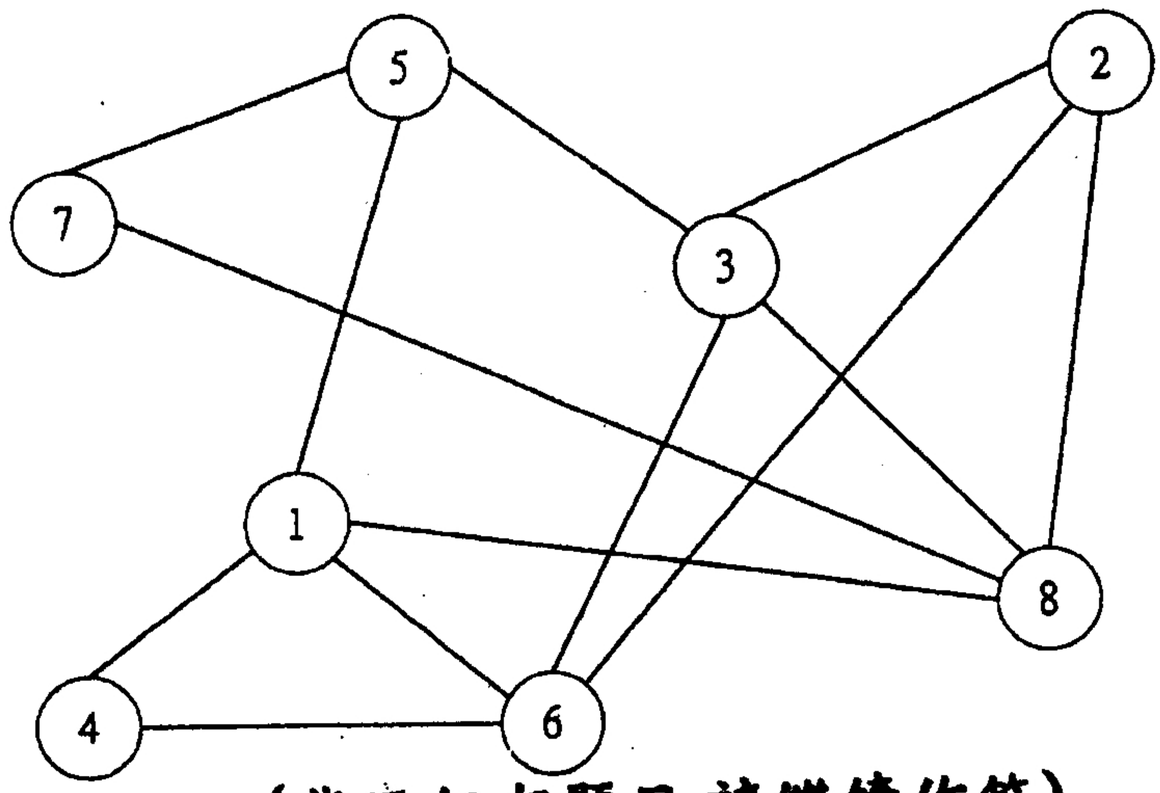
   of the same binary tree.

   (a) Draw a binary tree defined by such a pair of sequences. (10%)

   (b) Dose such a pair of sequences uniquely define a binary tree? (5%)

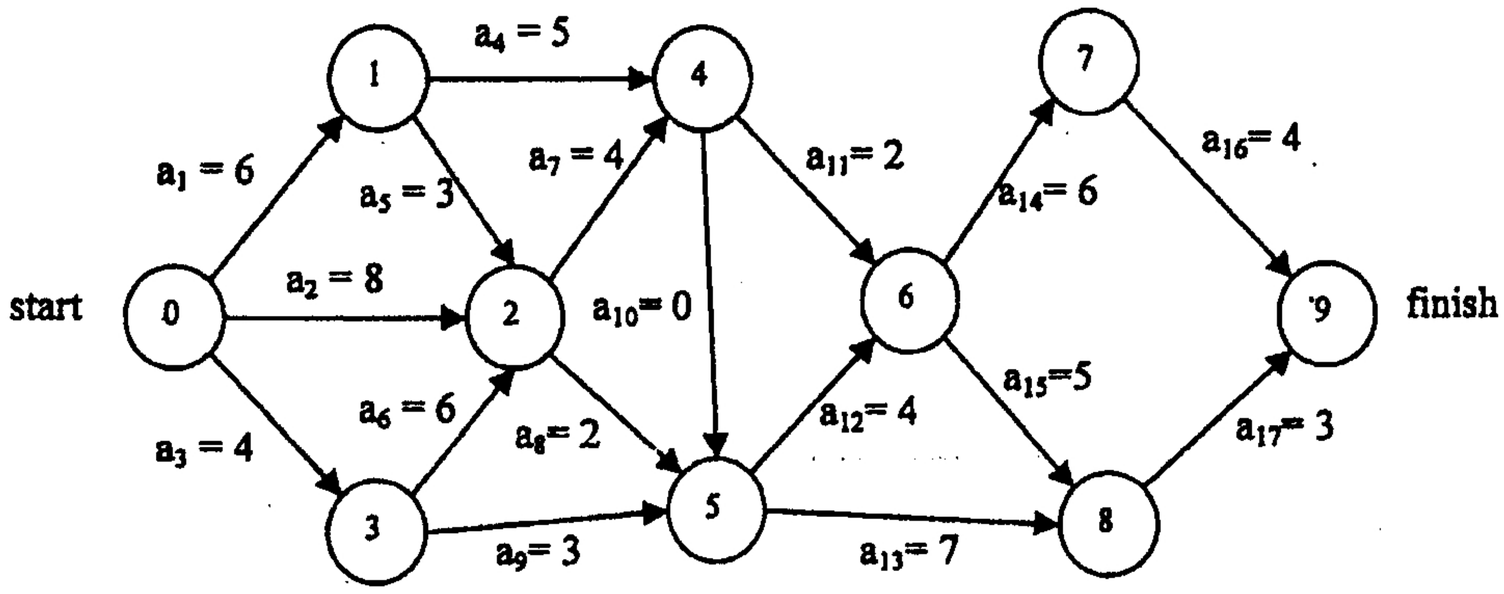
3. Under what conditions will the bubble sort run faster than the quick sort? Please state two conditions and explain the reason. (10%)

4. Use depth-first search and breadth-first search to traverse the following graph with node 1 as the starting node. Write down the depth-first and breadth-first traversal sequence. (At any moment, if more than one node can be visited next, always select the one with the smallest value to visit first.) (10%)



（背面仍有題目，請繼續作答）

5. The following directed graph is an AOE network which represents a project from its starting to its finishing. Compute the earliest time (ee), latest time (le), and the allowed slack of each activity. Then determine which activities are critical. (15%)



6. Given a set of messages and their weights in the following, construct the Huffman decoding tree that yields the minimum weight. (10%)

Message	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Weight	4	18	11	48	22	32	69	7	2	30

7. Given the following sequence of numbers:

29, 54, 31, 15, 67, 82, 8, 14, 93, 47

(a) If these numbers are inserted into an empty binary search tree sequentially, please draw the resultant binary search tree. (5%)

(b) If these numbers are inserted into an empty max heap sequentially, please draw the resultant max heap in its tree form. (5%)

8. In an $n \times n$ **N-matrix**, all terms other than those in column 1, column n, and the diagonal are zero. An N-matrix has at most $3n$-2 nonzero terms. An N-matrix can be compactly stored in a one-dimensional array by first storing column 1, then column n, and then the remaining elements of the main diagonal.

x 0 0 0 0 0 x
x x 0 0 0 0 x
x 0 x 0 0 0 x
x 0 0 x 0 0 x
x 0 0 0 x 0 x
x 0 0 0 0 x x
x 0 0 0 0 0 x

x denotes a nonzero number
all other terms are zero

Given the following data structures:

```
#define MAX_SIZE 1000     /* maximum size of the array */
typedef struct term {
        int row;
        int col;
        int value;
};
term nmatrix[MAX_SIZE];
bool set(term* nmatrix, int i, int j, int newValue);
```

Note that the row index and the column index of the matrix are both starting at 1 instead of 0. Please implement the function set() which stores newValue as the (i,j) element of the N-matrix, 1 <= i <= n and 1 <= j <= n. Note that Parameter nmatrix is a term pointer which points to the first element of the array which stores the matrix. The element is to be stored in the proper position of the one-dimensional array nmatrix. You can implement the function using C, or pseudo code. (15%)