所別：資訊管理學系碩士班 丙組 科目：資料結構

1. Let the recursive function Guess be defined as follows:

```
Unsigned Guess (unsigned X, unsigned Y)
  if (Y == 0) return 0;
  else if (Y == 1) return X;
  else
  {
    Z = Guess (X, Y/2);
    return Z*2 + X*(Y%2);
  }
```

   (a) Trace Guess (1, 10). (3%]
   (b) Describe the purpose of Guess. (3%)
   (c) Write down the recursive formula of function Guess. (4%)

2. Describe how one can use stacks and the push & pop operations to simulate an array's retrieval and storing operations, respectively. (10%)

3. Describe the purpose of the following code by tracing it on a list pointed by Head of at least 5 elements. (10%)

```
L = Head;
R = L->Next;
L->Next = NULL;
while (R != NULL)
{
  N = R->Next;
  R->Next = L;
  L = R;
  R = N;
};
Head = L;
```

4. Describe the data structure and algorithm for traversing a binary tree in level-order. (10%)

5. Add the following numbers one by one from left to right into an initially empty binary search tree.

   5 8 1 4 6 0 3 7 9 2

   (a) Draw the final binary search tree. (5%)
   (b) Draw the resulting binary search tree after deleting the root from the tree in (a). (5%)

注意：背面有試題

所別：**資訊管理學系碩士班 丙組** 科目：**資料結構**

6. Is Quick Sort always faster than Insertion Sort? Explain your answer. (5 %)

7. How many distinct topological orders are there in the following graph? (5 %)
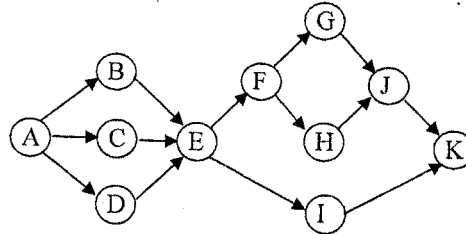
< Figure 1 >

8. The following questions are about hashing table.
   (a) What is collision? (2 %)
   (b) What is quadratic probing? (2%)
   (c) What is rehashing? (2%)
   (d) Does the perfect hashing exist? Explain your answer. (4%)

9. Suppose that smaller number node is visited first. According to the following graph,
   (e) Draw the depth-first spanning tree starting from node 1. (5 %)
   (f) Write the back edges of the depth-first spanning tree in (a). (5 %)
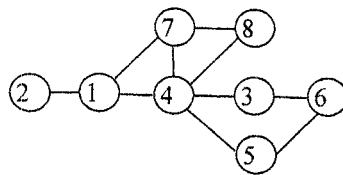
< Figure 2 >

10. A binary tree is stored in an array as follows: 《42、29、71、45、43、76、41、70》
    (a) Give the definition of a heap.(5 %)
    (b) Adjust the binary tree into a heap. You should draw the heaps for each step during the adjustment. (5 %)

11. Write an algorithm in pseudo-code to find the kth small element in an array. You are not allowed to presort the array. (10 %)