所別：資訊工程學系碩士班 科目：資料結構與演算法

1. (5%) Assuming that alphabetical order of key is used, show the result of inserting the following keys into a red-black tree:
   yes, no, maybe, true, false
   Be sure to mark a tree node as a red node or a black node.

2. Assume the edges of a graph are as follows in alphabetical order:
   S, T S,Z T,Y T,Z V,Y V,Z X,Y X,Z
   (a) (5%) Draw a sketch showing this graph.
   (b) (5%) Perform a depth-first iteration from S. Show visit order of the vertices.
   (c) (5%) Perform a breadth-first iteration from S. Show visit order of the vertices.

3. (5%) In open hashing, with the quotient-offset collision handler, insert the following keys into a table of size 11:
   20, 33, 49, 22, 26, 202, 140, 508, 9
   Show the result.

4. Please implement the stack by using an **array**.

```
#define MAX_STACK 100;
typedef int ITEM_TYPE;
typedef struct stack_type {
        ITEM_TYPE item[MAX_STACK];
        int       top;
} STACK_TYPE;
void create_stack (STACK_TYPE *stack); /* Make stack logically accessible */
```
(2%)

```
void destory_stack (STACK_TYPE *stack); /* Make stack logically
   inaccessible */
```
(3%)

```
BOOLEAN empty_stack (STACK_TYPE *stack); /* True if stack is empty */
```
(5%)

```
BOOLEAN full_stack (STACK_TYPE *stack); /* True if stack is full */
```
(5%)

```
void push (STACK_TYPE *stack, ITEM_TYPE new_item); /* Add item to the
   top of the stack */
```
(5%)

```
void pop (STACK_TYPE *stack, ITEM_TYPE *old_item); /* Remove item
   from the top of the stack */
```
(5%)

5. A string is a sequence of symbols; for example, $X = \langle x_1, x_2, \ldots, x_m \rangle$ is a string of m symbols $x_1, x_2, \ldots, x_m$. When we delete 0 or more symbols (not necessarily consecutive) from X, we get a subsequence of X.
   a) Write an algorithm using principle of optimality (dynamic programming) to

注意：背面有試題

calculate the length of the longest common subsequence of $X = \langle x_1, x_2, \ldots, x_m \rangle$ and $Y = \langle y_1, y_2, \ldots, y_n \rangle$. (15%)

b) What is principle of optimality? (10%)

6. For two points $P = (p_1, p_2)$ and $Q = (q_1, q_2)$ in the plane, we say that P ***dominates*** Q if $p_1 > q_1$ and $p_2 > q_2$. Given a set S of n points, the ***rank*** of a point P in S is the number of points in S dominated by P. The problem is to find the rank of every point in S. A straightforward way to solve this problem is to conduct an exhaustive comparison of all pairs of points. Hence, this approach requires $O(n^2)$ running time. Design a more efficient algorithm to solve this problem. Analyze the time efficiency of your algorithm. (15%)

7. How to merge k sorted lists with total length N efficiently. What is the execution time of your algorithm. (10%)