(100)輔仁大學碩士班招生考試試題	考試日期：100 年 3 月 18 日 第4 節
	本試題共 2 頁 (本頁為第 1 頁)

科目： 軟體設計　　系所組：資訊工程研究所

1. **(5 points)** Let A be an array of size $n \geq 6$ containing integers from 1 to $n-5$, inclusive, with exactly five repeated. Describe a fast algorithm with time complexity $O(n)$ for finding the five integers in A that are repeated.
2. **(5 points)** Describe in detail how to swap two nodes x and y (and not just their contents) in a singly linked list L given references only to x and y.
3. **(5 points)** Suppose you have two nonemepty stacks `S` and `T`, and a deque `D`. Describe how to use `D` so that `S` stores all the elements of `T` below all of its original elements, with both sets of elements still in their original order.
4. **(5 points)** When implementing the *breadth-first search* (*BFS*) algorithm for a given graph, will you use stack or queue to store the traversed vertices? Justify your answer.
5. **(10 points)** How do you evaluate that a *hash* function is good or not? What are the properties of *heap*?
6. **(20 points)** Consider the sequence of keys (5, 16, 22, 45, 2, 10, 18, 30, 50, 12, 1). Draw the result of inserting entries with these keys (in the given order) into
   (a) **(5 points)** An initially empty *binary search tree*.
   (b) **(5 points)** An initially empty *AVL tree*.
   (c) **(5 points)** An initially empty *(2, 4) tree*.
   (d) **(5 points)** An initially empty *red-black tree*.
7. **(5 points)** Show that $\sum_{k=1}^{n} \frac{\lg k}{k} = O(\lg^2 n)$.
8. **(9 points)** Give an asymptotic upper bound for the following recurrences.
   (a) **(4 points)** $T(n) = T(n-1) + 2n$.
   (b) **(5 points)** $T(n) = 2T(\frac{n}{2}) + \frac{n}{\lg n}$.
9. **(11 points)**
   (a) **(3 points)** What is the difference between two algorithm designing methods, divide-and-conquer and dynamic-programming?
   (b) **(8 points)** Show how to find the *minimum cost* while editing from a string "dog" to another string "cat". Only three operations, insert, delete, and exchange, are allowed to be used. The descriptions of three operations are shown as follows:
   "insert(x)" inserts a character x into a string.
   "delete(x)" removes a character x from a string.
   "exchange(x, y)" replaces a character x by a character y in a string.
   (Note: You can define any specific cost of each operation by yourself.)

※ 注意：1.考生須在「彌封答案卷」上作答。
2.本試題紙空白部份可當稿紙使用。
3.考生於作答時可否使用計算機、法典、字典或其他資料或工具，以簡章之規定為準。

10. **(8 points)** The following program is a sorting algorithm, called "`QuickSort`".

```
//"A" is an array of size n, A[1],A[2],A[3],…,A[n].
//"exchange(x,y)" is a procedure for exchanging data between x and y.

  void QuickSort(A)
  {for i = n downto 1
       {for j = 1 upto i-1
            if (A[j]>=A[j+1]) then
                 exchange(A[j],A[j+1])}}
```

(a) **(2 points)** Is above `QuickSort` algorithm a *stable* sorting algorithm? (why or why not?)

(b) **(6 points)** Analyze the growth order of above `QuickSort` algorithm running in the *average* case.

11. **(4 points)** A procedure used in the merge step of merge sort takes two input sorted lists and produces a single output sorted list by combining the items from both input lists. In merge sort, size of two given lists is approximately equal, but this procedure works correctly even when the lists are unequal in size. Suppose that we run this procedure on two sorted lists, one of length 10 and the other of length 30.

(a) How many comparisons this procedure uses in the best case?

(b) How many comparisons this procedure uses in the worst case?

(Note: Describe your reasons; otherwise you will get **zero**.)

12. **(5 points)** Given an integer `m` and a set `S` of `n` integers, please design an algorithm to determine whether there are two integers `x` and `y` in the set `S`, such that `x+y=m`.

(You will get **5 points**, if you justify your design is an `O(nlogn)` algorithm.)

(You will get **2 points**, if you justify your design is an $O(n^2)$ algorithm.)

13. **(8 points)** (True/False) (Note: Describe your reasons; otherwise you will get **zero**.)

(a) **(2 points)** G is a weighted directed graph with a *negative cycle* and then there is no pair of vertices in G has a shortest path.

(b) **(2 points)** G is a weighted directed graph with a *negative edge* and then every shortest path (from a single source) computed by *Dijkstra's* algorithm is wrong.

(c) **(2 points)** G is a weighted directed graph. Every shortest path (from a single source) in G won't change if an extra weight is added on every edge of G.

(d) **(2 points)** Every weighted undirected graph G has a *unique* minimum spanning tree.