Q.V

國立中央大學94學年度碩士班考試入學試題卷　共 2 頁　第 1 頁

所別：資訊管理學系碩士班 丙組 科目：資料結構

1 (20 points) Write a Java or C++ program that *completely* inverts an arbitrary non-recursive list *l*. For example, if *l=(a,(b,c))*, then *inverse(l) = ((c,b),a)*. The list *l* has no shared sub-list, and may contain sub-list. Your program needs to declare the required data structure. You only need to provide the *inverse* method/function, i.e., you don't need to deal with input and output.
// Correctness of the program (15 points). Comments of the program (5 points)

2 (10 points) Let us represent a maze of dimension $m \times p$ by a two-dimensional array, $maze[i][j]$, where $1 \le i \le m$ and $1 \le j \le p$. A value of 1 implies a blocked path and a 0 means one can walk right on through. You can choose any of 8 directions from a block to next block. The length of a path means the number of blocks this path covers. What is the maximum path length from start to finish for any maze of dimension $m \times p$? How will the location of starting and finishing points affect your answer?

3 (5 points) What is the maximum number of nodes in a k-ary tree of height h?

4 (15 points) *Level-order traversal* for a binary tree is a traversal method that visits the root first, then the root's left child, followed by the root's right child. We continue in this manner, visiting all the nodes at each new level from the leftmost node to the right most node. Please write a non-recursive traversal method of *level-order traversal* for binary tree. Your program needs to declare any required data structure, and you only need to write the key method/functions in Java or C++. You may assume the existence of common supporting classes except tree (e.g., array, stack, queue, and list). You can also assume the methods such as *add* and *delete* are available for the supporting class.
// Correctness of the program (12 points). Comments of the program (3 points)

注意：背面有試題

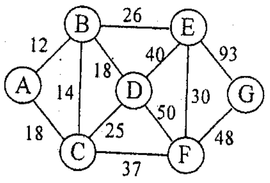
0.14

國立中央大學94學年度碩士班考試入學試題卷

所別：資訊管理學系碩士班 丙組 科目：資料結構

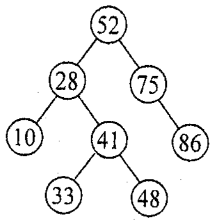
5. 當我們進行 Hashing 運算時，似乎並沒有一種 Hashing Function 可以完全避免碰撞(Collision)或溢位(overflow)，請說明碰撞和溢位有何不同？(5 分)

6. Prim's Method and Kruskal's Method 可用來建構花費最少擴張樹(Minimum Cost Spanning Tree)，請根據圖一請回答下列問題：(每小題 5 分)
   (a) 以 Prim's Method 找出該圖的花費最少擴張樹，須繪出每一步驟的變化情形。
   (b) 以 Kruskal's Method 找出該圖的花費最少擴張樹，須繪出每一步驟的變化情形。



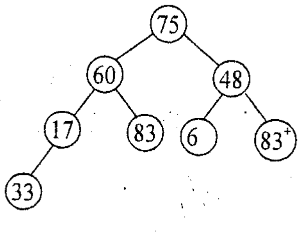
< 圖一 >

7. 圖二爲一高度平衡樹(Height Balanced Binary Tree)，又稱 AVL Tree：
   (a) 請在該樹中加入 36，其結果必須仍爲 AVL Tree。(10 分)
   (b) 承(a)，請在(a)的結果中加入 50，其結果必須仍爲 AVL Tree。(10 分)



< 圖二 >

8. 請針對 Heap Sort 回答下列問題
   (a) 是否爲穩定的排序(stable sorting)？(5 分)
   (b) 有一組未排序的資料爲《75、60、48、17、83、6、83$^+$、98》，以陣列將其建成二元樹如圖三，試以 Heap Sort 對其進行排序，請圖示每一步驟後，Heap Tree 的變化情形。(10 分)



< 圖三 >