

國立中央大學 110 學年度碩士班考試入學試題

所別： 資訊管理暨大數據分析類

共 3 頁 第 1 頁

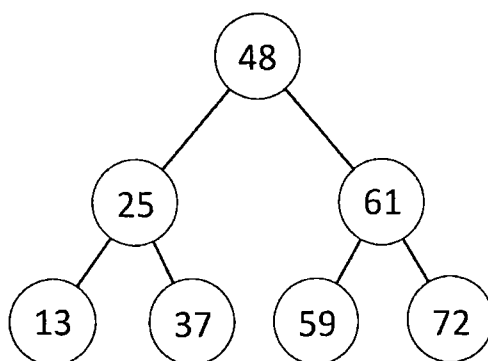
科目： 資料結構

本科考試禁用計算器

*請在答案卷(卡)內作答

【題型說明：除第 2(b)題為證明題外，其餘皆依題意為問答題或程式設計題】

1. Please process the given data { 71, 66, 2, 18, 9, 23, 17, 61, 52, 81 } from small to large according to the specified method of each question.
 - (a) Please show the process in detail and the result of **Quick Sort** (use pivot). (6%)
 - (b) Please show the process in detail and the result of **Bubble Sort**. (6%)
 - (c) Please show the process in detail and the result of **Heap Sort**. (6%)
 - (d) Please use the hash function $h(m)=m\%11$ and linear probing collision resolution to create a hash table with 11 elements. Then, show this **hash table**. (7%)
2. Let T denote a 2-3 tree. The height of T is h.
 - (a) What is the maximum number of leaves ? (1%)
 - (b) Prove your answer to (a). (8%)
3. Link list is one of the popular data structures.
 - (a) Please briefly explain linked list (3%) and doubly linked list (3%) by drawing their structures.
 - (b) Please use C code (ANSI C) to implement corresponding functions for inserting node at the head of linked list(4%) and doubly linked list(5%).
4. Binary tree is another popular data structure.
 - (a) According to the binary tree shown below, please provide final results of pre-order (3%), in-order (3%), and post-order (3%) traversals.
 - (b) As mentioned above, please use C code (ANSI C) to implement corresponding functions for printing visited elements by using pre-order(3%), in-order (3%), and post-order (3%).



注意:背面有試題

國立中央大學 110 學年度碩士班考試入學試題

所別： 資訊管理暨大數據分析類

共3頁 第2頁

科目： 資料結構

本科考試禁用計算器

*請在答案卷(卡)內作答

5. This question concerns with the Depth First Search (DFS) or Depth First Traversal algorithm.

(a) It is common to implement the DFS algorithm with a Stack. However, the stack may be used differently in different implementations. You are tasked to implement the DFS algorithm for any given graphs. Explain how you are going to use the stack in your implementation. (7%)

(b) Next, write down the pseudocode for your DFS algorithm following implementation specifications below. You must include your idea given in (a) in your pseudocode. You may include any additional functions required for your implementation. Note: your pseudocode should be clearly written so that any programmer is able to read and implement your pseudocode. (20%)

```
Global visited = [] //array for visited vertices, you don't have to use it
                        //depending on your implementation.
```

```
Global s = new Stack() //the stack object
```

```
Global G = new Graph() //the graph object
```

```
//Both array and stack have standard operations: push() and pop()
```

```
//In addition, stack has top() operation which points to the top element in the stack.
```

```
//Graph.adj(vx) returns a list of vertices adjacent to the given vertex vx
```

```
//in ascending order
```

```
//The DFS algorithm, vx is the starting vertex.
```

```
DFS(G, vx) {
```

```
}
```

注意:背面有試題

國立中央大學 110 學年度碩士班考試入學試題

所別： 資訊管理暨大數據分析類

共 3 頁 第 3 頁

科目： 資料結構

本科考試禁用計算器

*請在答案卷(卡)內作答

(c) Refer to the pseudocode below for this question. Given a graph G , use your pseudocode in part (b) above to traverse G and print the order in which the vertices are visited. Your answer should be printed like this: (6%)

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

$V = \{0,1,2,3,4,5\}$

$E = \{(0,1),(0,2),(0,3),(1,4),(2,3),(2,5),(3,4),(4,5)\}$

$G = \{V, E\}$

$vx = V[0]$

$DFS(G,vx)$