淡江大學 105 學年度碩士班招生考試試題

系別：資訊工程學系 B 組　　科目：資料結構

考試日期：3 月 5 日(星期六) 第 2 節　　本試題共 8 大題， 2 頁

本試題雙面印刷

1. (12%) Find the complexity of the following algorithms by big O analysis.
   (a) Determining all duplicates in a singly link list.
   (b) Determines all possible links between n vertices.
   (c) Count the number of operations where *statement* is executed.

```
for (int i = 0; i < n; i +=2)
    for (int j = 1; j < n; j++)
        {statement;}
```

   (d) Count the number of operations where *statement* is executed.

```
for (int i = 0; i < n; i *=2)
    for (int j = 1; j < n; j++)
        {statement;}
```

2. (32%) Consider the following data structure

```
typedef struct node {
    struct node *next;
    struct node *prev;
}tNode;
typedef struct header {
    struct node *head;
    struct node *tail;
} tHeader;
```

   (a) Assume a pointer "tHeader *h". Please allocate memory space for *h.* (We say the list maintained by *h* is the list L).
   (b) Allocate and then insert a new node "tNode *n" to L
   (c) Allocate and then insert another new node "tNode *n" to the head of L
   (d) Allocate and then insert another new node "tNode *n" to the tail of L without a loop.
   (e) Allocate and then insert another new node "tNode *n" behind the second node in L
   (f) Remove and then free the last node in L without a loop
   (g) Remove and then free the updated last node in L without a loop
   (h) Remove and then free the first node in L
3. (6%) Consider the following function

```
int foo(int a, int b) {
    if (a%b == 0) return b;
    else return foo(b, a%b);
}
```

   (a) What is the output given by foo(13, 3)?
   (b) Briefly explain the purpose of the foo function.

背面尚有試題

淡江大學 105 學年度碩士班招生考試試題

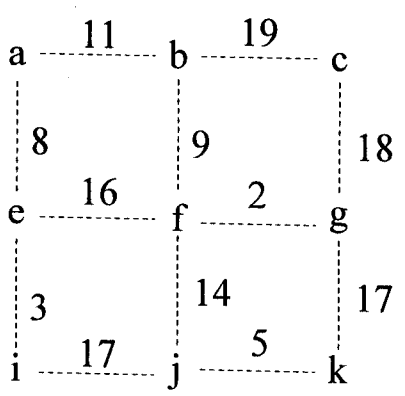
系別：資訊工程學系 B 組　　科目：資料結構

考試日期：3 月 5 日(星期六) 第 2 節　　本試題共 8 大題， 2 頁

4. (10%) Suppose you were asked to write a method that will take two sorted stacks A and B (min on top), e.g., "Stack A = new stack" and create one stack that is sorted (min on top), e.g., "Stack C = new stack". You are allowed to use only the stack operations such as pop, push, size and top, e.g., A.pop(), A.push(), A.size(), A.top(). Please fill the following function.

```
Public Stack mergeSortedStacks(Stack A, Stack B) {
    Stack C = new stack;
    {fill the following code segment}}
```

5. (20%) In the following questions, consider list of numbers: 62, 31, 70, 91, 25, 11, 9, 61, 73, 6.
   (a) Show the result of inserting the numbers in the list in the same order specified above into an initially empty minimum heap. Note that you need to show how the heap looks like after each number is inserted.
   (b) Show the result of inserting the numbers in the list in the same order specified above into an initially empty binary search tree. Note that you need to show how the binary search tree looks like after each number is inserted.
   (c) Use the binary search tree you created in question 5(b). What are the two possible binary search trees after 62 is deleted?
   (d) Explain how you can utilize a minimum heap to sort the list of numbers in descending order. Let n be the number of elements in the list. What is the time complexity of your sorting algorithm?
6. (5%) Order the following functions by their growth rate from slowest to fastest: N, $N^{0.5}$, N log log N, 2/N, 2^N , N log(N^2)
7. (5%) Identify a minimal spanning tree (MST) in the following graph using Kruskal's algorithm. Write down the edges in the MST in the order in which Kruskal's algorithm adds them to the tree and make those edges into solid lines. Also write down the total weight of the minimum spanning tree that you have identified.



8. (10%) Write a recursive function with the prototype int isHeap(node *root); The function returns 1 if the complete binary tree (with the given root) is heap, otherwise it returns 0. (You can use root->left and root->right to indicate left and right structures of the root structure, respectively. You can also use root->element, root->left->element, root->right->element to specify the element in the structure)