國立嘉義大學99學年度

資訊管理學系碩士班（乙組）招生考試試題

科目：資料結構

1. 請回答以下的程式問題：

   (1) 分析程式執行效能時，可以直接插入系統函式（例如 C 的 time( )）來紀錄程式的開始與結束時間，但此方法的正確性受到很多因素影響，請列出五點可能因素、並簡單解釋。 (10 Points)

   (2) 已知 C 語言變數宣告：**int a, b, c;**

   請問 **if(!(a<=100) || !(a>200)) b = c;** 這個程式片段有何特點？會造成哪些問題？ (5 Points)

   (3) Singly linked list 可以用於實作 Stack，例如右邊的 push 函式。若不考慮一些錯誤檢查等問題，畫出從空的 Stack（假設為 s）連續 PUSH ***A, B, C*** 三個 item 之後的 Chain。並說明這個 Stack 結構的缺點是什麼？ (5 Points)

```
void push(stackPointer *top, element item){
  stackPointer *temp;
  temp=(stack *)malloc(sizeof(stack));
  temp->data = item; temp->link=NULL;
  if( *top != NULL) (*top)->link = temp;
  (*top) = temp;
  }
```

2. For a digraph ***G*** with 3 vertices and 3 weighted edges, the vertex ***A*** is adjacent from vertex ***B*** that the edge length is 2, the length of <***A***, ***C***> is 1, and the vertex ***A*** is adjacent to ***B*** that the edge length is -1. Please draw the graph ***G***. And, does the shortest path from the vertex ***B*** to vertex ***C*** exist? If yes, give the length. Otherwise, explain the reason. (5 Points)

3. Consider a binary search tree with the input sequence (**C, A, F, G, D, B, E**).

   (1) Draw the binary search tree after inserting the entire sequence. (5 Points)

   (2) Draw both possible binary trees after deleting **F**. Which is better? Why? (10 Points)

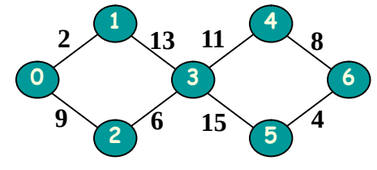
4. A recursive Fibonacci function is defined as：

```
int F(int n) {
        if( n == 0 ) return 0;
        if( n == 1 ) return 1;
        if( n == 2 ) return 2;
        return (F(n-3) - F(n-2) + F(n-1));
        }
```

   (1) How many times is the function F() called for F(6) (including the initially invoked F(6))? (5 Points)

   (2) Please analysis the space complexity of the F function. (5 Points)

5. The following graph is a classic network topology represented by an undirected graph with weighted edges. Please answer the following questions. (NOTE: In order to get a unique answer, you MUST convert the graph to an adjacency list in which the adjacent nodes are linked by the increasing order.)



   (1) Draw its DFS(6) spanning tree (please mark the visited order from 0 to 6). And, how do we apply such trees in the real communication networks? (5 Points)

   (2) Draw its minimum spanning tree with the Sollin's Algorithm. (5 Points)

   (3) Is it a biconnected graph? Why? What does it mean in the real communication network environment? (5 Points)

背面尚有試題

6. Please answer the following sorting problems.

(1) Which statements are correct?

(A) The higher order of merging, the less computing time we will have.

(B) Quick sort has the best average behavior and is unstable.

(C) Merge sort is good for sorting huge amount of data.

(D) Insertion sort is best for small amount of data since its code is simple and needs only O(*1*) extra space.

(E) Table sort can be used to avoid excessive data movement while the amount of data of each record to be sorted is large. (5 Points)

(2) The quick sort algorithm is used to sort the input list {37, 20, 12, 62, 43, 56, 34, 72, 15}. Support that we pick the median of the first, middle, and last keys in each sublist as the pivot. What is the pivot key now? And write down the status of the list after the first pass. (5 Points)

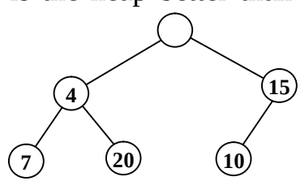
(3) Which sorting methods: insertion sort, quick sort, merge sort, radix sort, heap sort, and bubble sort are stable? Why is “stable sorting” a good property? (5 Points)

7. Answer the following problems of priority queues (PQs).

(1) What is a priority queue? Give an example that computer systems use PQ. (5 Points)

(2) To implement single-ended PQs, why is the heap better than the sorted array structure? (5 Points)

(3) The deap structure can be used to implement double-ended priority queues. Is this a legal **deap** structure? Why? (5 Points)



8. Actually, you can use either AOV or AOE network to represent a project planning. Please discuss the difference between both methods. (5 Points)