注意事項

一、不可於試題紙上作答。

二、When you are asked to derive the time complexity of a program (or algorithm) in the form of $O(f(n))$, the function $f(n)$ should be expressed in the ***simplest*** and ***tightest*** form. For example, if running time $T(n)=2n^2+3$, you should write $T(n)=O(n^2)$. In other words, you will get 0 points if you write $T(n)=O(n^3)$ or $T(n)=O(2n^2+3)$.

1. (a) [5 points] Given the size n of the input data, where n is a positive integer, we assume that the running time of a program is $O(f(n))$. State the *formal* definition of $O(f(n))$.

   (b) [5 points] Given the size n of the input data, where n is a positive integer, we assume that the program requires the running time $T(n)=O(f(n))$, where

$$T(n)=\log 1+\log 2+\log 3+\cdots+\log n=\sum_{x=1}^{n}\log x.$$

   Derive the function $f(n)$.

2. [10 points] What would be the contents of queue `Q` after the following pseudo code is executed? The contents of the input file are `8, 16, 15, 4, 0, 4, 6, 8, 17, 24, 0, 33, 47, 28, 9, 0`. Note that your answer should point out where is the front of the queue.

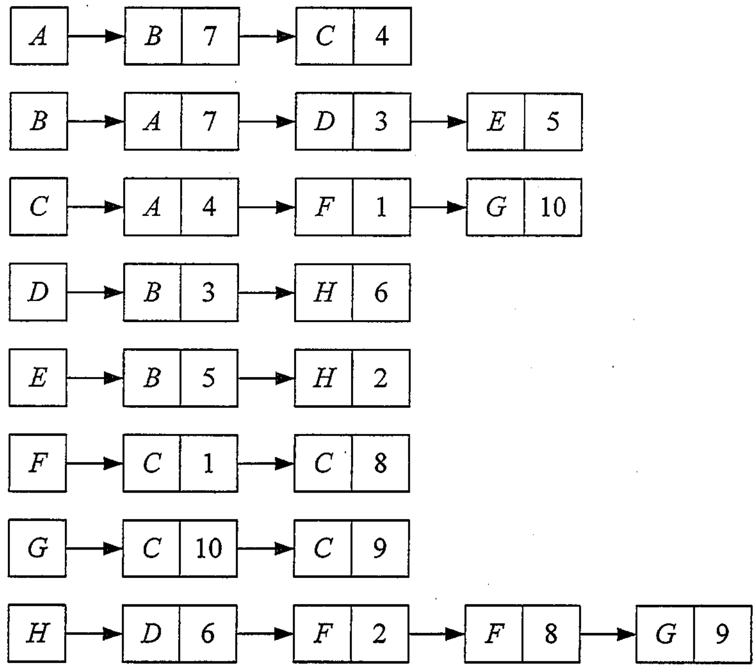
```
Q = create_Queue();
S = create_Stack();
while (not end of file) {
   x = read_an_input_number();
   if ( x != 0 )
      push_Stack(S, x);
   else {
      x = pop_Stack(S);
      x = pop_Stack(S);
      while ( S is not empty ) {
          x = pop_Stack(S);
          add_Queue(Q, x);
          // Add the integer x into the queue Q.
      }
      delete_Queue(Q);
      // Delete one element from the queue Q.
   }
}
```

國立中山大學 101 學年度碩士暨碩士專班招生考試試題

科目：資料結構【電機系碩士班丙組選考】

題號：4062
共 4 頁 第 2 頁

3. On the basis of the adjacency list shown below, answer the following questions. Note that the value field in the element of the adjacency list contains the edge weight of the graph.



Vertex list Adjacency list

(a) [5 points] Draw the graph *G* whose adjacency list is defined above.
(b) [5 points] Temporarily ignore the edge weight and use the DFS (depth first search) to traverse the graph *G* from the vertex *A*.
(c) [5 points] Temporarily ignore the edge weight and use the BFS (breadth first search) to traverse the graph *G* from the vertex *A*.
(d) [10 points] Star from the vertex *A* and use Prim's algorithm to find the minimum cost spanning tree of the graph *G*. Show the actions step by step.

4. Insert a sequence of keys `{7, 9, 16, 30, 49, 82, 5, 33, 31, 6, 2, 1}`, in that order, into a data structure which has no keys initially.
(a) [5 points] Construct a binary search tree for that sequence.
(b) [5 points] Construct a max heap for that sequence.
(c) [5 points] Tell us the resultant max heap after deleting `9` and `49` from the max heap obtained in question 4(b).

5. (a) [5 points] Given an unsorted integer array of size n, does the binary search algorithm outperform the sequential search algorithm? Use the big-O notation to justify your answer.

(b) [10 points] Given an integer array **A** of size n, the following pseudo code shows the insertion sort algorithm. Now, suppose that we want to sort an integer array **B** of size 100. In other words, the array **B** contains 100 integers. Derive the worst case running time of `insertion_sort(B)` in terms of big-O notation. (Note that you will get 0 points if you just give the answer directly.)

```
void insertion_sort(array A) {
  int i, j, key;

  for (j = 2; j <= array_size(A); j++) {
     key = A[j];
     // Insert A[j] into the sorted sequence A[1..j-1].
     i = j-1;
     while ( (i > 0) && (A[i] > key) ) {
        A[i+1] = A[i];
        i = i-1;
     }
     A[i+1] = key;
  }
}
```

6. The sequence L_n of Lucas numbers is defined as follows.

$$L_n = \begin{cases} 2 & \text{if } n = 0, \\ 1 & \text{if } n = 1, \\ L_{n-1} + L_{n-2} & \text{if } n \geq 2. \end{cases}$$

The LUCAS NUMBER PROBLEM is defined as "Given an integer $n \geq 0$, output the n-th Lucas number L_n." The following recursive function `Lucas(int n)` can solve the LUCAS NUMBER PROBLEM.

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

(a) [10 points] Prove that, when $n \geq 2$, the running time $T(n)$ of `Lucas(n)` is larger than $\left(\frac{1+\sqrt{5}}{2}\right)^{n-2}$. Note that the running time is measured in units of instruction step.

(b) [5 points] Now, tell us whether the LUCAS NUMBER PROBLEM is NP-complete? Explain your reasons. (Note that you will get 0 points if you do not present any reasons.)

7. [10 points] Given the preorder sequence, *A B D H J E C F G*, and the inorder sequence, *H D J B E A F C G*, please (1) draw the corresponding binary tree, and (2) show its postorder sequence.