※ 考生請注意：本試題 □可 ☑不可 使用計算機

1. Please use heap tree to sort the 5, 2, 4, 7, 3, 6, 3 in increasing order. (10%)
 (a) Should a min or max heap tree be used? (2%)
 (b) Please discuss how this tree is constructed. (4%)
 (c) What will the tree be if 10 is inserted into this it? (4%)

2. Assume you input an infix equation "((a+b)*c+d-e)/(f+g)-(h+j)*(k-l)/(m-n)" in a program, which will be transferred to different orders for processing.
 A. Please draw this equation in an inorder binary tree (5%)
 B. Please show the postorder of the tree (5%)
 C. If an equation is with parentheses " ( ) ", explain which data structure could be used to find parentheses matching and how (please use the equation in question A equation as an example). (5%)
 D. Suppose there are three orders of a tree, named preorder, postorder, and inorder. If you can only choose two of these three orders, (i.e. preorder & postorder), please list two combinations which can form a precise tree structure? (5%)

3. Please define what min leftist tree is? (4%) The following figure shows two min leftist trees. Please describe in detail how to meld them into a tree. (6%)

4. A template class is defined as below and the current condition in memory is shown as the figure. Please write down a pseudo code to complete the following process:
 A. Insert a node between 2nd "b" & 3rd "c" nodes. (5%)
 B. Delete first node (please remember to clear unused memory) (5%)

```
template <class T>
    class ChainNode
    {
      private:
      T data;
      ChainNode<T> *link;
      ChainNode() {}     //Constructor 1
      ChainNode(const T& data)   //Constructor 2
         {this->data = data;}
    };
    first = new ChainNode <char> ( ); …..//most codes are abbreviated
```


first

（背面仍有題目,請繼續作答）

※ 考生請注意：本試題 □可 ☑不可 使用計算機

5. (28%) True or False, and EXPLAIN (4% each)

Answer **T** or **F** for each of the following statements to indicate whether the statement is true or false, respectively. If the statement is correct, briefly state why. If the statement is wrong, explain why or give a counter example. Answers without reasons will get <u>at most 1</u> point.

(a) (**T** , **F**) Given two integers r and k. To calculate r^k takes $\Omega(k)$ time.

(b) (**T** , **F**) Given an array $a[i]$ of $n + 1$ real numbers for $i = 0, ..., n$. To calculate $f(r) = \sum_{i=0}^{n} a_i r^i$ takes $\Omega(n \log n)$ time.

(c) (**T** , **F**) Given a tree $T = (V, E)$, suppose each undirected edge $(i, j) \in E$ has a negative length c_{ij}. Then, the path from u to v identified by Dijkstra's algorithm will still be the shortest path.

(d) (**T** , **F**) Given a rooted tree $T = (V, E)$, the preorder traversal and postorder traversal will always give different outputs.

(e) (**T** , **F**) If we store n numbers by a binary search tree, it takes $O(n)$ time to sort these numbers in nonincreasing order.

(f) (**T** , **F**) To identify the 5th largest number from an array of $n \in [10^4, 10^{20}]$ takes $O(1)$ time.

(g) (**T** , **F**) In a red-black tree of n nodes that store n distinct values, it takes $O(\log n)$ time to find the median (i.e. the $\frac{n}{2}$ largest) of these numbers.

6. (22%) In the undirected connected graph below, each vertex represents a street intersection, each edge represents a road section and the weight gives the average time needed to travel that road section. Answer the following questions:

(a) [4%] Suppose you want to find the route from intersection A to intersection B that minimizes the <u>number of intersections</u>. Described how you would solve this problem in general for a graph of n vertices and m edges (i.e. give an algorithm which is applicable not just for this specific case). You should also discuss the complexity of your method.

(b) [4%] based on your answer in (a), demonstrate your solution for the given graph. Show your work to enough detail that we can see how you reached your solution.

(c) [4%] Suppose you want to find the route from intersection A to intersection B that minimizes the <u>total travel time it takes</u>. Described how you would solve this problem in general for a graph of n vertices and m edges (i.e. give an algorithm which is applicable not just for this specific case). You should also discuss the complexity of your method.

(d) [4%] based on your answer in (a), demonstrate your solution for the given graph. Show your work to enough detail that we can see how you reached your solution.

(e) [6%] Suppose you are asked to assign direction for each edge in the graph to make it a digraph such that each vertex can connect to each other vertex by some directed path (i.e. strongly connected). How do you know whether such strongly connected orientation exists for an undirected connected graph G of n vertices and m edges? Explain your method and discuss its complexity.