1. Give a weighted graph. Please write the steps of deriving minimum cost spanning trees by Kruskal’s and Prim’s algorithms. (20%)

2. The node structure of a linked list is defined as:
   typedef struct Node *NodePtr;
   typedef struct Node {
       NodePtr next;
       int data;
   };

   Suppose there is a linked list, pointed by the pointer \( p \). The linked list has at least one node. Please write an algorithm to reverse the linked list. (10%)

3. Show that quicksort’s best-case running time is \( \Omega(n \log n) \). (15%)

4. What are the minimum and maximum numbers of elements in a heap of height \( h \)? (15%)

5. A recursive algorithm for computing the size of a tree can be:
   ```c
   int size(Node n) {
       if (x == NULL) return 0;
       return 1 + size(n.leftChild) + size(n.rightChild);
   }
   ```

   Please write an iterative algorithm for computing the size of a tree. A stack data structure with push( ) and pop( ) operations can be assumed. (10%)

6. Assume a matrix \( data[m][n] \) is used to store integral data. Also, assume integers in a same row or same column are in increasing order. Please (a) describe an algorithm to search an integer \( num \) in \( data[ ][ ] \); (b) analyze time complexity of your algorithm. (10%)

   | 1 2 3 |
   | 5 6 19 |
   | 7 8 20 |
   | 11 12 23 |

7. Given an algorithm below, please determine the values in \( f[ ] \) when \( pat = \text{“abcababaca”} \). Note that variable declaration is omitted in the code. (10%)

   ```c
   n = strlen(pat);
   f[0] = -1;
   for (j=1; j < n; j++) {
       i = f[j-1];
       while ((pat[j] != pat[i+1]) && (i >= 0))
           i = f[i];
       if (pat[j] == pat[i+1])
           f[j] = i+1;
       else f[j] = -1;
   }
   ```

8. Prove or disapprove the following statements. (10%)
   (a) \( \sum_{i=1}^{n} i^3 = \Theta(n^4) \)
   (b) \( n^2 + 6n^3 = O(n^3) \)