1. The ***Fibonacci number*** can be defined in a recurrence relation: (20%)
   F(n) = F(n-1) + F(n-2) for n >= 2 with F(0) = F(1) = 1.
   a) Write a recursive program for computing F(n), and analyze the complexity of the program.
   b) Write a non-recursive program for computing F(n) in liner time.
2. Using **C++ (or C)** code to implement the following functions for the **Doubly Linked Circular Lists with head node.** (20%)
   a) Write the C++ Class (or C Struct) definition of a Doubly Linked Circular list.
   b) Write the (member) function ***insertright*** which inserts a node with information field ***x*** to the right of a node pointed to by a pointer ***p*** in a double linked circular list.
   c) Write the (member) function ***deletenode*** which the node pointed to by a pointer ***p*** from a doubly linked circular listed and stores its contents in ***x***.
3. We divide the list to be sorted into two roughly equal parts called the left and the right sublists. The sublists are sorted recursively, and the sorted sublists are merged. (20%)
   a) Write a **Recursive Merge Sort** algorithm.
   b) Write the status of the list L = (25, 6, 80, 1, 58, 10, 59, 16, 49, 18) at the end of each phase of the Recursive Merge Sort.
   c) Is the resulting function a stable function? And what is its computing time (Big O)?
4. **Prim**'s algorithm and **Sollin**'s algorithm: (20%)
   a) Using Prim's algorithm and Sollin's algorithm find the minimum-cost spanning tree of Figure 1.
   b) Write the Prim's algorithm.
   c) Write the Sollin's algorithm.
   d) Discuss the difference of this two algorithms and its time complexity.
5. Explain the following terms: (20%)
   a) Priority queue.
   b) Huffman codes.
   c) Max-Min Heap.
   d) Hash tables.
   e) Binary search tree.

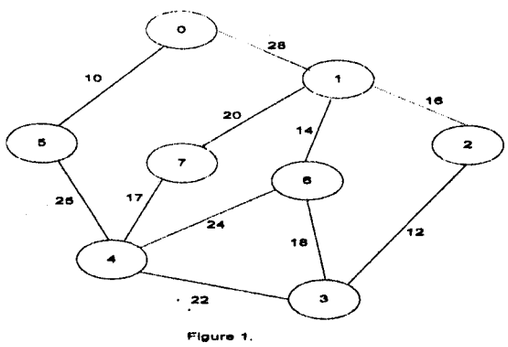

Figure 1.