1. Write the C code to implement the following functions for the complex number abstract data type; the data structure is

```
typedef struct{
    double real;
    double imaginary;
} complex;
```

(a) Subtract_complex(), which takes the difference of two complex numbers.
void subtract_complex(complex *p_difference, complex *p_complex1, complex *p_complex2); (5%)
(b) Equal_complex(), which compares the equality of two complex numbers. (5%)
(c) Multiply_complex(), which multiplies two complex numbers. (5%)

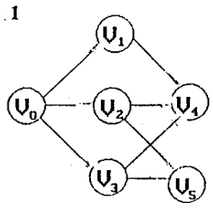
2. Assume that a binary search tree is represent as a threaded binary search tree.
(a) Insert the key values: 7,16,49,82,5,31,6,2 into the tree and explain the algorithem you used. (5%)
(b) Write a function to search an element. (5%)
(c) Write a function to delete an element. (5%)
(d) What is the time complexity of your function for (b) and (c). (5%)

3. (a) Explain what is a priority queue , use two methods to implement the priority queue with data structure. (5%)
(b) Suppose you use a HEAP to implement this queue , explain how to do the insert and delete functions. (10%)
(c) Show the time complexity (Big O) of your priority queue for insert and delete. (5%)

4. Under what circumstances would you recommend the use of each of the following sorts over the others? Why?
(a) Shell sort. (5%)
(b) Heapsort. (5%)
(c) Quicksort. (5%)

5. For the graph shown in fig. 1,
(a) Show the adjacency Matrix and its adjacency lists of this graph. (5%)
(b) Describe the Breadth First Search (BFS) algorithm to traversal of this graph starts at vertex v_0 . (10%)

Figure 1



6. For the digraph shown in fig.2,
(a) Write an algorithm SHORTEST-PATHS to obtain in nondecreasing order the lengths of the shortest paths from vertex 1 to all remaining vertices in the digraph. (10%)
(b) List the shortest paths from vertex 1 to all remaining vertices in this digraph. (5%)

Figure 2