Course Title: Algorithms Design & Analysis

Course Code: CSE - 225

Question Patterns

Q.1 **a.** Suppose we find the 8th term using the recursive implementation. The arguments passed to the function calls will be as follows and justify the answer:

```
fibonacci(8)
fibonacci(7) + fibonacci(6)
fibonacci(6) + fibonacci(5) + fibonacci(5) + fibonacci(4)
fibonacci(5) + fibonacci(4) + fibonacci(4) + fibonacci(3) +
fibonacci(4) + fibonacci(3)
            + fibonacci(3) + fibonacci(2)
:
:
:
```

Which property is shown by the above function calls?

a) Memoization
b) Optimal substructure
c) Overlapping sub-problems
d) Greedy

b. Solve the following instance of the 0/1 knapsack problem using dynamic programming approach. Assume that the knapsack capacity is 7.

Item	**Weight**	**Benefit**
1	3	$5
2	4	$6
3	5	$7
4	6	$8

c. Using substitution method proves that the running time of Merge sort algorithm is O(nlogn).

d. Deduce the best case complexity of Quick sort.

Q.2 **a.** Find an optimal solution to the fractional knapsack problem, where the knapsack capacity is M = 13.

Item	**Weight**	**Profit**
1	**2**	**10**
2	**3**	**5**
3	**5**	**15**
4	**7**	**7**

b. What is the advantage of Huffman's Greedy algorithm in data compression? Suppose A,B,C,D,E,F,G and H are 8 items and suppose they are assigned weights as follows:

Data items:	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Weight :	**22**	**5**	**11**	**19**	**2**	**11**	**25**	**5**

Now construct the Tree by using Huffman Algorithm and encoding the each node.

c. Determine the space complexity of following code:

```
int sum(int a[], int n)
 {
  int r = 0;
  for (int i = 0; i < n; ++i)
  {
    r += a[i];
  }
   return r;
  }
```

Q.3 **a.** Explain the following statement with correct answer:

The main time taking step in fractional knapsack problem is ___________

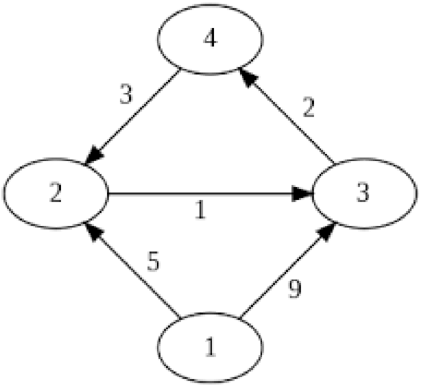
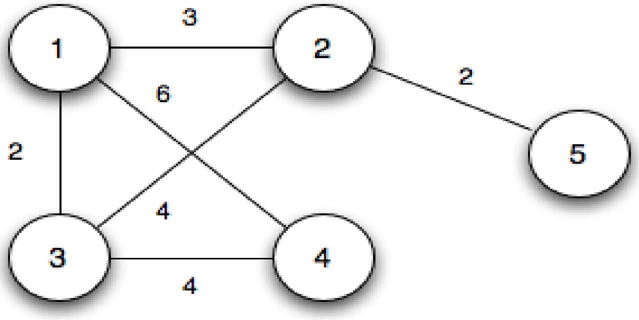
a) Breaking items into fraction

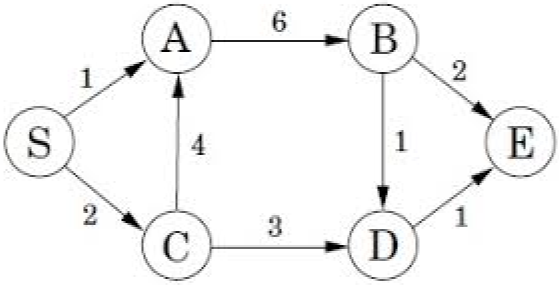
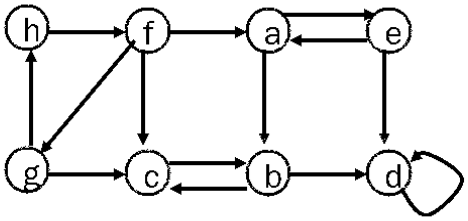
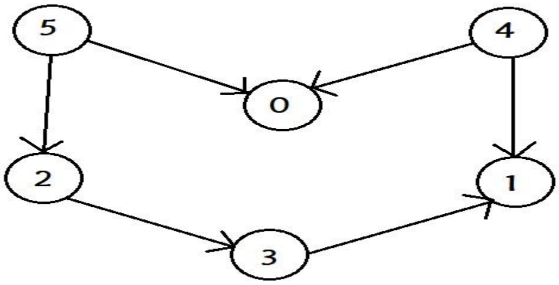
b) Adding items into knapsack

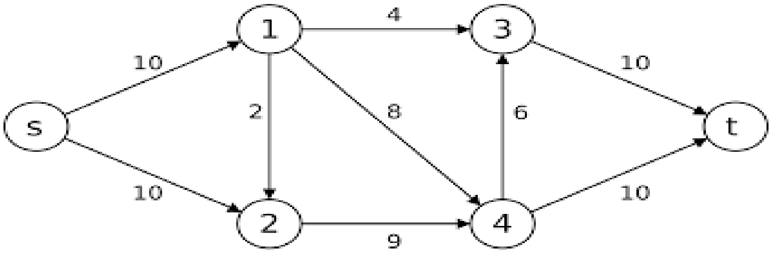
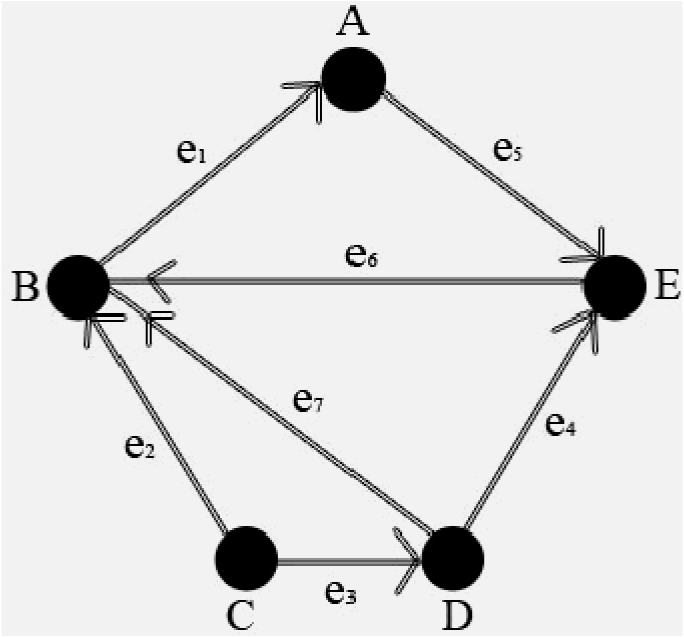
c) Sorting

d) Looping through sorted items

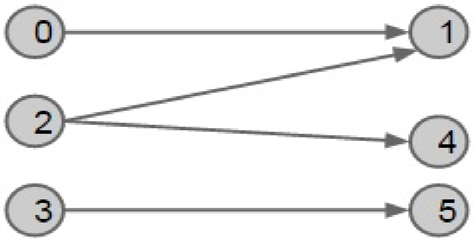
b. Prove that Longest Common Subsequence for input Sequences **"ABCD"** and **"AED"** is **"AD"** of length 2.

	c.	Apply Floyd-Warshall algorithm to find all-pairs shortest-paths of the graph *F* given in Fig. 1.  **Fig. 1: A weighted graph *F***
Q.4	**a.**	Prove that for minimum spanning tree, E=V-1; E is the number of edge of minimum spanning tree and V is the number of nodes of graph from minimum spanning tree has been found.
	b.	Apply Prim's algorithm on the weighted graph *P* given in Fig. 3 to find a minimum spanning tree of *P*.  **Fig. 2: A weighted graph *P***
	c.	Apply Kruskal's algorithm on the weighted graph *P* given in Fig. 2 to find a minimum spanning tree of *P*.

Q.5	**a.**	Given the directed graph *G* in Fig. 3, find a shortest path from source **S** to each of the vertices using Dijkstra algorithm.  A B S E C D 6 1 2 4 1 2 3 1 **Fig. 3: A directed graph G**
	b.	What is negative cycle? Which one is more appropriate single source shortest path algorithm (Dijkstra or Bellman-Ford) to find out the shortest path from graph with negative edge? Justify.
	c.	Find the strongly connected components of the directed graph S given in Fig. 4. Then draw the component graph of S  **Fig. 4: A directed graph S**
Q.6	**a.**	Apply Depth First Search Algorithm on the directed graph P in Fig. 5 by using Topological Sort.  **Fig. 5: A directed graph *P***

	b.	Justify that time complexity of Breadth First Search Tree is O(V+E), where V is the number of vertices of graph and E is the number of edges of graph.
	c.	Define branch-and-bound. Write a backtracking algorithm to color a map with no more than four colors
Q.7	**a.**	Define NP, NP hard and NP Complete.
	b.	Apply Ford Fulkerson Method for Max Flow Problem on the graph F in Fig. 6  s 1 2 3 4 t 10 10 4 2 8 6 9 10 10 **Fig. 6: A Directed graph *F***
	c.	Find out the Hamiltonian Cycle of following graph:  A B C D E e1 e2 e3 e4 e5 e6 e7
Q.8	**a.**	Consider the following factorial code : #include <stdio.h> // Iterative function to find factorial of a number using for loop

		unsigned long factorial(int n)
		{
		    unsigned long fact = 1;
		    int i;
		    for (i = 1; i <= n; i++)
		        fact = fact * i;
		    return fact;
		}
		// Program to find factorial of a number
		int main()
		{
		    int n = 5;
		    printf("The Factorial of %d is %lu", n, factorial(n));
		
		    return 0;
		}
		What type of dynamic programming it is (Bottom up or top down)? Also clarify the reason.

	b.	Apply Bipartite Graph to assign job Y= {1,4,5} to X={0,2,3} on Graph . 
	c.	What is the basic distinction between greedy method and dynamic method?
Q.9	**a.**	Apply DFS by using topological sorting on the graph F in Fig. 6

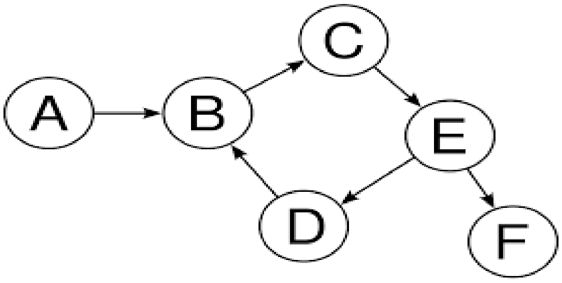
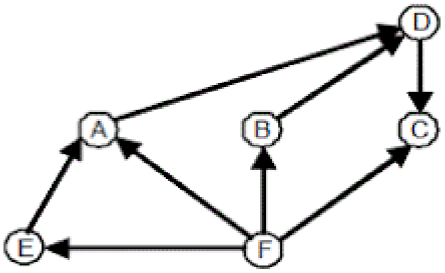


Fig. 6: A directed graph *F*

b. What would be the time complexity of following code:

```
#include <stdio.h>
int main()
{
    int n, i, flag = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    for(i = 2; i <= n/2; ++i)
    {
        // condition for nonprime number
        if(n%i == 0)
        {
            flag = 1;
            break;
        }
    }
    if (n == 1)
    {
      printf("1 is neither a prime nor a composite
           number.");
    }
    else
    {
        if (flag == 0)
          printf("%d is a prime number.", n);
        else
          printf("%d is not a prime number.", n);
    }
     return 0;
}
```

	c.	What are the disadvantages of recursion?
10.	**a.**	Apply Topological sorting on graph P in Fig. 5  **Fig. 5: A directed graph *P***
	b.	Using tree recursion method proves that the running time of Quick sort algorithm is O(nlogn) in best case; where n is the number of input.
	c.	Define the worst case and best case of Depth first search and also determine the complexity.