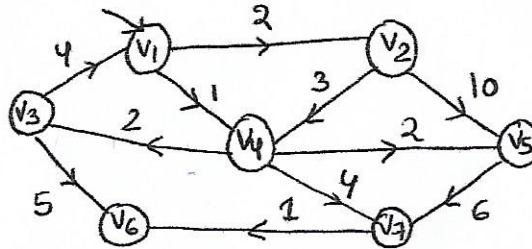


**Minor Test**  
**MCA301: Design and Analysis of Algorithms**

**Time: 1 hr.**

**Max Marks: 20**

1. (a) Use Dijkstra's shortest path algorithm to obtain shortest path from the vertex  $v_1$  to all remaining vertices in the following graph. Show step by step generation of shortest path. Write the path to be taken from  $v_1$  to reach other vertices. (3)
- (b) Treating the following graph as undirected graph, apply prim's algorithm to generate minimum cost spanning tree and compute the minimum cost. (3)



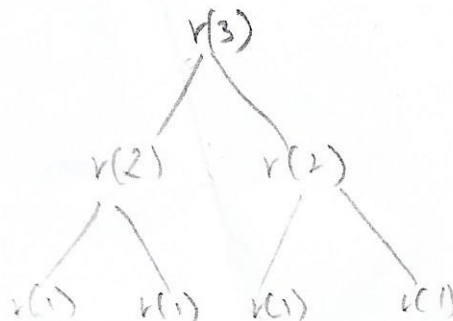
2. Analyze the worst case time complexity of heap sort. (3)
- (b) Write non-recursive version of quicksort. Apply your algorithm to the following input and establish the fact that quicksort is not a stable sorting algorithm. (5)

80, 100, 40, 30, 80, 75

3. Find the time complexity of the following C function (assume  $n > 0$ ). Use either recursion tree or back substitution. (3)

```
int recursive (int n)
{
    if (n==0)
        return 1;
    else
        return (recursive(n-1)+ recursive(n-1));
}
```

4. Given a set  $S$  of  $n$  integers, write an  $O(n \log n)$  time algorithm that determines, for a given integer  $x$ , whether there is a pair of elements in  $S$  whose sum is  $x$ . Give the step by step analysis in 2-3 lines. (3)



$n = n - 1$   
 $1 + x = n$   
 $n = 1 + x$   
 $k = n - 1$   
 $n - k = 1$   
  
 $1 - n$   
 $n = 1 + k$   
 $n - 1 = k$   
 $n - k = 1$   
 $-k = 1 - n$   
  
 $k =$   
  
 $T(n) = T(n-1) + T(n-1) + c$   
 $= 2T(n-1) + c$   
 $= 2T(n-1)$