

1721



Page No: 73

Date: / /

Q7 Prim's algorithm

```
#include <stdio.h>
```

```
int a, b, u, t, m, n, i, j;
```

```
int v[10], e = 1, min = 0; cost[10][10];
```

```
int main() {
```

```
    printf("Enter no. of vertices: ");
```

```
    scanf("%d", &m);
```

```
    printf("Enter matrix: ");
```

```
    for (i = 1; i <= m; i++) {
```

```
        for (j = 1; j <= m; j++) {
```

```
            scanf("%d", &cost[i][j]);
```

```
            if (cost[i][j] == 0)
```

```
                cost[i][j] = 999;
```

```
        }
```

```
    v[1] = 1;
```

```
    while (e < m) {
```

```
        for (i = 1, min = 999; i <= m; i++) {
```

```
            for (j = 1; j <= m; j++) {
```

```
                if (cost[i][j] < min) { if (v[i] != 0)
```

```
                    { min = cost[i][j];
```

```
                      a = u = i;
```

```
                      b = t = j; } }
```

```
        }
```

```
        if (v[u] == 0 || v[t] == 0) {
```

```
            printf("Edge (%d) : (%d %d)\n",
```

```
                cost[i][j], e++, a, b, m);
```



Page No. 34

Date: / /

```
min = m;
```

```
v[b] = 1;
```

```
}
```

```
cost[a][b] = cost[b][a] = 999;
```

```
}
```

```
printf("Min cost = %.d", min);
```

```
return 0;
```

```
}
```




Q-7) Kruskal algorithm

```
#include <stdio.h>
```

```
int i, j, k, a, b, u, v, n, ne = 1;
```

```
int min, mincost = 0, cost[10][10],
```

```
parent[10];
```

```
int main() {
```

```
    printf("Enter vertices :- ");
```

```
    scanf("%d", &n);
```

```
    printf("Enter adjacency matrix: ");
```

```
    for (i = 1; i <= n; i++) {
```

```
        for (j = 1; j <= n; j++) {
```

```
            scanf("%d", &cost[i][j]);
```

```
            if (cost[i][j] == 0)
```

```
                cost[i][j] = 999;
```

```
        }
```

```
    } while (ne < n) {
```

```
        for (i = 1; min = 999; i <= n; i++) {
```

```
            for (j = 1; j <= n; j++) {
```

```
                if (cost[i][j] < min) {
```

```
                    min = cost[i][j];
```

```
                    a = u = i;
```

```
                    b = v = j; } }
```

```
        while (parent[u])
```

```
            u = parent[u];
```

```
        while (parent[v])
```

```
            v = parent[v];
```



```
if (u != v) {  
    printf ("Edge (%d, %d) : (%d, %d, %d)\n", u, v, a, b, min);  
    cost += min;  
    parent[v] = u; }  
cost[a][b] = cost[b][a] = 999;  
}  
printf ("Minimum Cost = %d", mincost);  
return 0;  
}
```