Dijistra's Algo . . -

Amber Mishra
1BM18CS013

```
class Dijk
{
    static int No_parent = -1;

    static void dijsktra ( int ar[] [], int src)
    {
        int n = ar[0].length;

        int dist [] = new int [n];
        boolean v [] = new boolean [n];
        for (int i = 0; i < n; i++)
        {
            dist[i] = Integer. MAX_Value;

        }
        dist[src] = 0;
        int parent [] = new int [n];

        parent [src] = No_parent;
        for (int i = 1; i < n; i++)
        {      int c = -1;      int min = Integer.MAX_VALUE;

            for (int j = 0; j < n; j++)
            {
```

```
if (! visited[j] && dist [j] < min )
  {
       r = j;
     min = dist [j];
   }
}.
visited [r] = true;

for (int j=0; j <n; j++)
  {
     int edgedist = ar[r][j];
     if ( edgedist >0 && (min+edgedist ) < dist [j]))
       {
         ~~taed~~ parent[j] = r;
         dist [j] = min + edgedist.
       }
   }
}.
void main ()
{
    Scanner x = new Scanner (System . in);
     int n= x. nextInt();
     int ar[] [] = new int [n][n];
```

```
for (int i =0; i<n; i++)
 {
    for (int j =0; j<n; j++)
     {   ar[i][j]= x.nextInt();
      }
   }
  int k= x.nextInt();
  dijstra (ar,k);
 }
}.
```