DATE: PAGE:

Algorithm:

```
dijkstra (a[max][max], int n, int sn)
 {
  int cost[max][max], distance[max], pred[max];
  int visited[max], count, mindist, next_node, i, j;
  for (i=0; i<n; i++)
   for (j=0; j<n; j++)
      if (a[i][j]==0)
          cost[i][j]=infinity;
      else
          cost[i][j]=a[i][j];
  for (i=0; i<n; i++){
        distance[i] = cost[sn][i];
        pred[i]=sn;
        visited[i]=0;
   }
  distance[sn]=0;
  visited[sn]=1;
  count=1;
  while(count<n-1){
      mindist = infinity;
      for(i=0; i<n; i++)
         if(distance[i] < mindist ~~&&~~ !visited[i])
           {
             mindist = distance[i];
             nextnode = i;
         }
      visited[next_node]=1;
       for (i=0; i<n; i++)
if (!visited [i])
    if (mindist + cost [next_node][i] < distance [i]) {
      distance [i] = mindist + cost [next_node][i];
        pred [i] = next_node;
    }
  count++;
}
for (i=0; i<n; i++)
    if (i != sn)
      {
      cout << "distance of node" << i << "is:" << distance [i];
      cout << "\nPath" << i;
      j = i;
      do
      {
       j = pred [j];
      cout << "<-" << j;
     } while (j != sn);
   }
  }
```