Shortest Path Lenght: it's an interesting metric, because given two nodes it tells us the minimum number of links we need to traverse.

If we have weighted links, I may be interested in finding the shortest path lenght not only in the number of links to be traversed, but in the minimum weight (for example) in doing so

If we consider an oriented graph, we use the Dijkstra algorithm

In each iteration, the group of N total nodes is divided in:

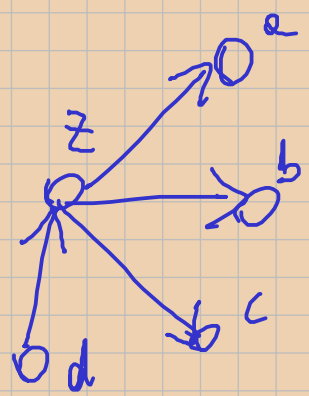
- V, visited nodes
- F, next to the visited nodes
- S, nodes to be yet examinated

$N = F + V + S$ in the initial step

1) We start from a node z, with d_z minimum ($d_z = 0$)

2) We move the z node from F to V

3) We move all successors of z in F



here a,b,c are successors, d is not

4) for every successor w of z, we update the valor d_w and u_w (they will be explained later)

$$d_w = \min \{ d_w, \; d_z + \rho_a \}$$

ρ_a → distance between dz

If the value of d_w was actually modified, we make: $u_w = z$

the name of the first node

$$U = \{ u_1, u_2, \ldots, u_n \}$$

We do this for every successor of z

5) we go to the node (not z) that has the minimum d (We cannot go to a node that was already visited)

If some nodes have the same d, we must make a choice