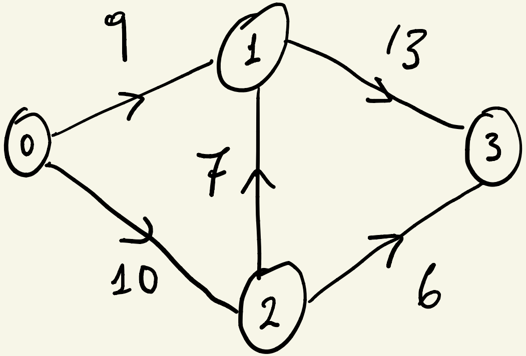

node label

$$G = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 9 & 10 & 0 \\ 1 & 0 & 0 & 0 & 13 \\ 2 & 0 & 7 & 0 & 6 \\ 3 & 0 & 0 & 0 & 0 \end{array}$$

row is from
column is to

— = neighbours of row number

source ↓

0	1	2	3
∞	∞	∞	∞
0	∞	∞	∞
0	9	∞	∞
0	9	∞	22

visited = [0, 1, 3]

unvisited = [2]
↓
∞

dijkstras input will be source label s, G

list of lists

```
N = len(G)
tentative_distance = [np.inf]*N
nums = np.arange(0, N)
visited = [ ]
t_d[s] = 0.

current = s
while len(visited) != N :

    d_d = { }
```

smallest t_d among unvisited is ∞

```
for n in nums:
    if G[current][n] != 0 and n not in visited:
        if t_d[n] > t_d[current] + G[current][n]:
            t_d[n] = t_d[current] + G[current][n]
        d_d[t_d[n]] = n

visited.append(current)
current = d_d[min(d_d.keys())]
min_unvisited = np.inf
for n in nums:
    if n in visited:
        continue
    else:
        min_unvisited = t_d[n]
```

```
                break
            if min_unvisited == np.inf:
                break

    current = neighbours[0][0]
    tmp = neighbours[0][1]
    for j in range(len(neighbours)):
        if neighbours[j][1] < tmp:
            current = neighbours[j][0]
            tmp = neighbours[j][1]
```

	0	1	2	3	4	5	6
0	○						
1		○					
2			○				
3				○			
4					○		
5						○	
6							○