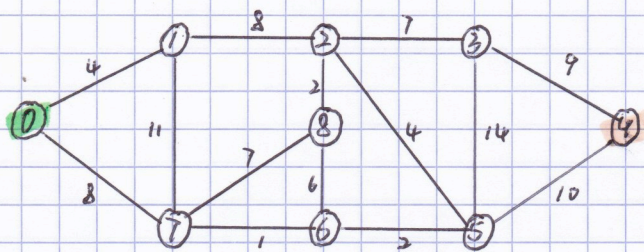


Dijkstra 最短路径算法



步骤:

1. 每次从未标记的节点中选择距离出发点最近的节点, 标记, 收录到最优路径集合中.

2. 计算刚加入节点A到邻近节点B的距离, (不包含已标记节点), 若 (节点A的距离 + 节点A到节点B的距离) < 节点B的距离, 则

① → ② → ⑦ = 4 + 8 = 12 < 15
① → ⑦ = 8 < 15
更新节点B的距离和前面点pre.

更新节点B的距离和前面点pre.

3. 直到标记到目标节点

Put "0":

	0	1	2	3	4	5	6	7	8
dist	0	4	∞	∞	∞	∞	∞	8	∞
pre	-	0	-	-	-	-	-	0	-

Put "1":

	0	1	2	3	4	5	6	7	8
dist	0	4	12	∞	∞	∞	∞	15	∞
pre	-	0	1	-	-	-	-	0	-

Put "7":

	0	1	2	3	4	5	6	7	8
dist	0	4	12	∞	∞	∞	9	8	15
pre	-	0	1	-	-	-	7	0	7

Put "6":

	0	1	2	3	4	5	6	7	8
dist	0	4	12	∞	∞	11	9	8	15
pre	-	0	1	-	-	6	7	0	7

Put "5":

	0	1	2	3	4	5	6	7	8
dist	0	4	12	25	21	11	9	8	15
pre	-	0	1	5	5	6	7	0	7

Put "2":

	0	1	2	3	4	5	6	7	8
dist	0	4	12	19	21	11	9	8	14
pre	-	0	1	2	5	6	7	0	2

Put "8":

	0	1	2	3	4	5	6	7	8
dist	0	4	12	19	21	11	9	8	14
pre	-	0	1	2	5	6	7	0	2

Put "3":

	0	1	2	3	4	5	6	7	8
dist	0	4	12	19	21	11	9	8	14
pre	-	0	1	2	5	6	7	0	2

Put "4":

	0	1	2	3	4	5	6	7	8
dist	0	4	12	19	21	11	9	8	14
pre	-	0	1	2	5	6	7	0	2

Shortest distance
最短路径: 21

⇒ 最短路径: shortest path
0 → 7 → 6 → 5 → 4