

# Dijkstra算法实现



主讲人：邓哲也



# Dijkstra算法实现

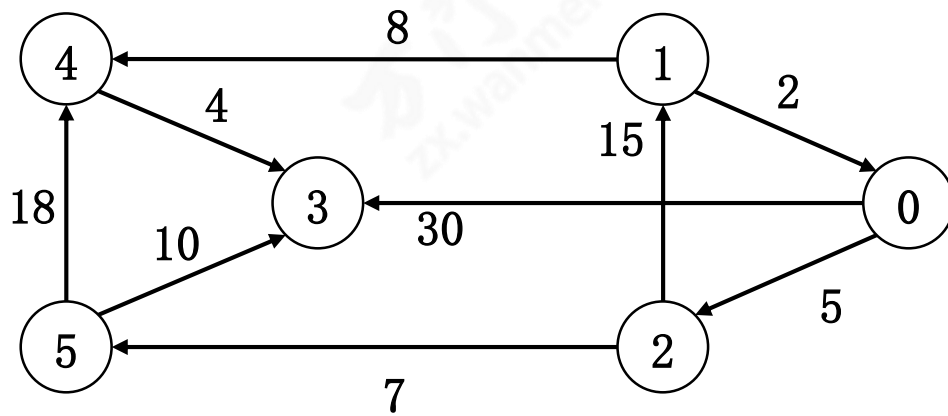
- 在 Dijkstra 算法里，为了求源点  $v_0$  到其他各顶点  $v_i$  的最短路径及长度，需要设置 3 个数组：
  - **dist[i]**: 表示当前找到的从源点  $v_0$  到终点  $v_i$  的最短路径的长度，初始时， $\text{dist}[i] = \text{edge}[v_0][i]$
  - **S[i]**: 表示顶点  $v_i$  是否加入到集合 S 中，1 表示已加入。初始时， $S[v_0]$  为 1，其余为 0.
  - **path[i]**: 表示  $v_0$  到  $v_i$  的最短路径上顶点  $v_i$  的前一个顶点序号。通过这个数组可以确定  $v_0$  到  $v_i$  的最短路径上的每一个顶点。

# Dijkstra算法实现

- 在 Dijkstra 算法里，重复做以下 3 步工作：
  1. 在数组  $\text{dist}[i]$  里查找  $S[i] \neq 1$ ，并且  $\text{dist}[i]$  最小的顶点  $u$ 。
  2. 将  $S[u]$  改为 1，表示顶点  $u$  已经加入进来了。
  3. 修改  $T$  集合中每个顶点  $v_k$  的  $\text{dist}$  及  $\text{path}$  数组值。

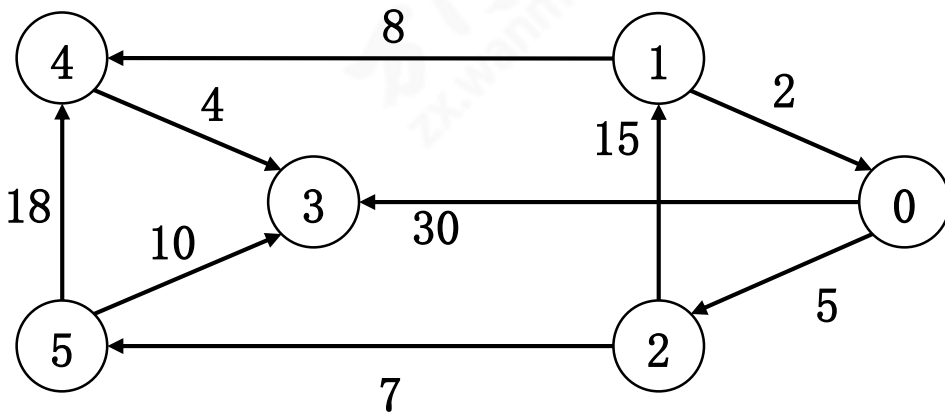
# Dijkstra算法实现

- 让我们回到这张图，看看这个算法是怎么运行的。



# Dijkstra算法实现

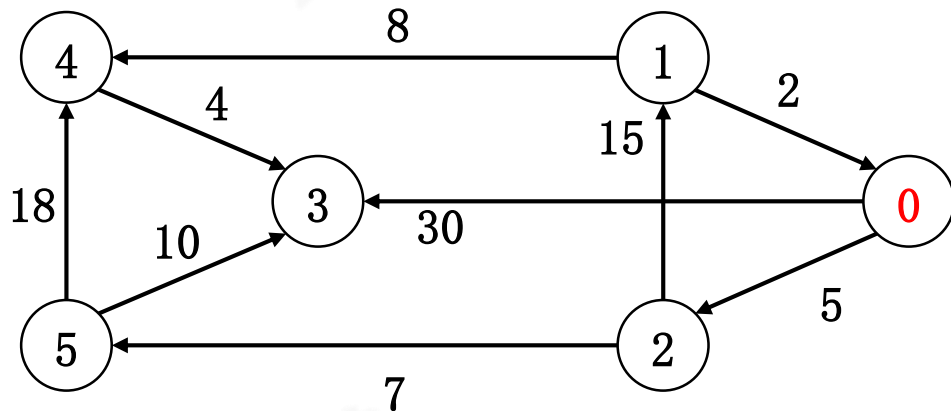
- 初始状态时,  $S[0]$  为 1, 其余为 0, 表示初始时,  $S$  中集合只有源点  $v_0$ , 其他顶点都在  $T$  中。



	0	1	2	3	4	5
S	1	0	0	0	0	0
dist	0	$\infty$	5	30	$\infty$	$\infty$
path	-1	-1	0	0	-1	-1

# Dijkstra算法实现

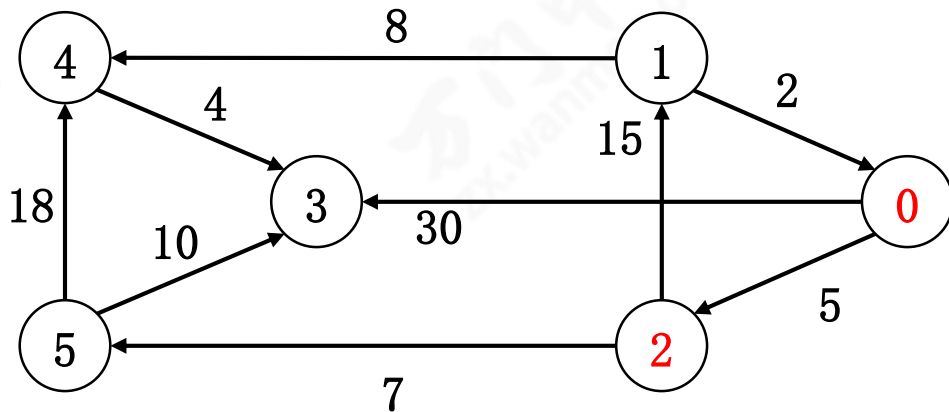
- `dist` 数组中各元素的初始值就是  $v_0$  到它的直接路径。
- `path[2]` 和 `path[3]` 为 0，表示当前求得的 0 到 2 的最短路和 0 到 3 的最短路中，前一个顶点是 0. -1 表示 0 到其余顶点没有直接路径。



	0	1	2	3	4	5
S	1	0	0	0	0	0
dist	0	$\infty$	5	30	$\infty$	$\infty$
path	-1	-1	0	0	-1	-1

# Dijkstra算法实现

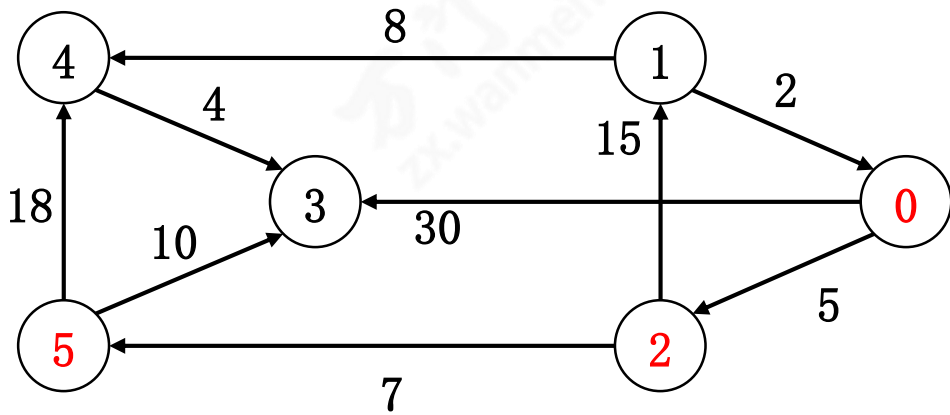
- 求出顶点 2 的最短路径，对三个数组作出修改。



	0	1	2	3	4	5
S	1	0	<u>1</u>	0	0	0
dist	0	<u>20</u>	5	30	$\infty$	<u>12</u>
path	-1	<u>2</u>	0	0	-1	<u>2</u>

# Dijkstra算法实现

- 求出顶点 5 的最短路径，对三个数组作出修改。

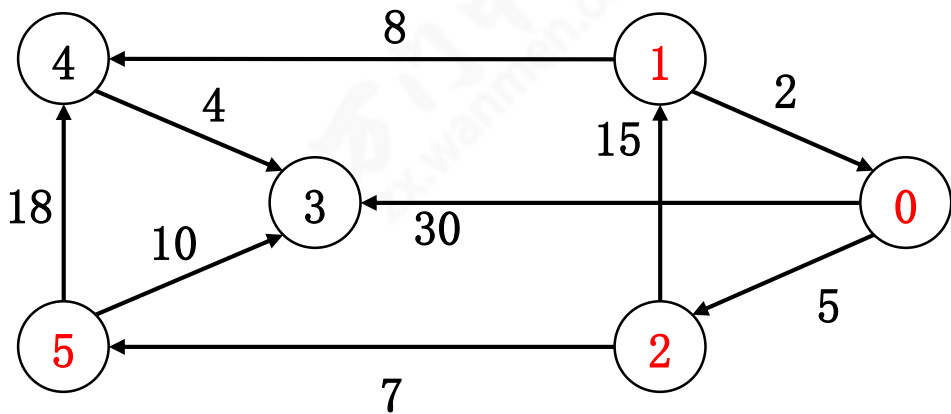


	0	1	2	3	4	5
S	1	0	1	0	0	<u>1</u>
dist	0	20	5	<u>22</u>	<u>30</u>	12
path	-1	2	0	<u>5</u>	<u>5</u>	2



# Dijkstra算法实现

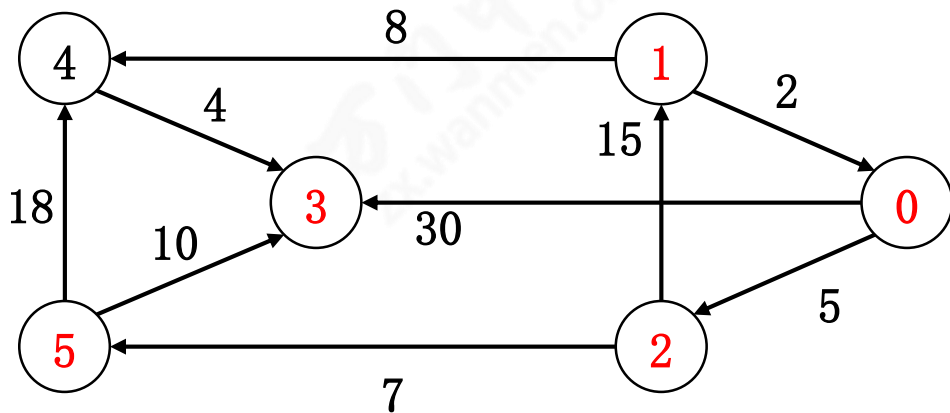
- 求出顶点 1 的最短路径，对三个数组作出修改。



	0	1	2	3	4	5
S	1	<u>1</u>	1	0	0	1
dist	0	20	5	22	<u>28</u>	12
path	-1	2	0	5	<u>1</u>	2

# Dijkstra算法实现

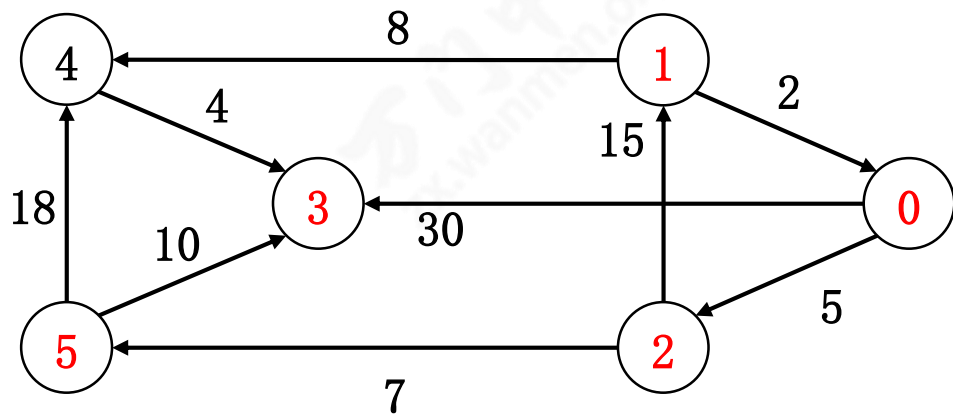
- 求出顶点 3 的最短路径，对三个数组作出修改。



	0	1	2	3	4	5
S	1	1	1	1	0	1
dist	0	20	5	22	28	12
path	-1	2	0	5	1	2

# Dijkstra算法实现

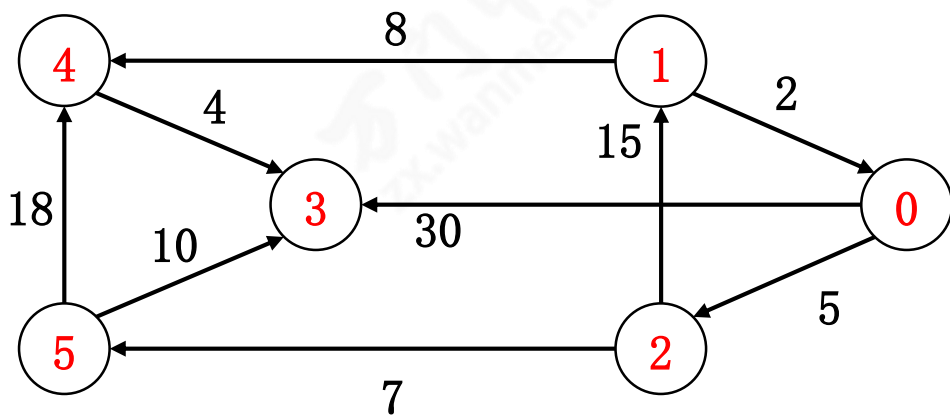
- 求出顶点 3 的最短路径，对三个数组作出修改。



	0	1	2	3	4	5
S	1	1	1	<u>1</u>	0	1
dist	0	20	5	22	28	12
path	-1	2	0	5	1	2

# Dijkstra算法实现

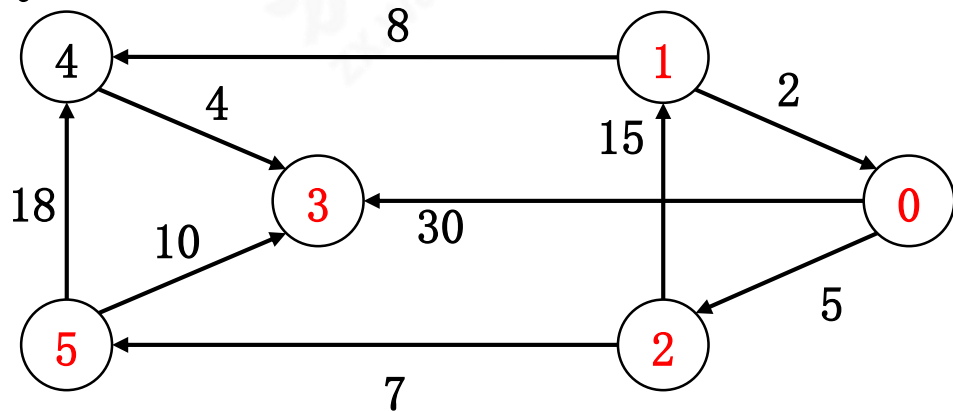
- 求出顶点 4 的最短路径，对三个数组作出修改。



	0	1	2	3	4	5
S	1	1	1	1	<u>1</u>	1
dist	0	20	5	22	28	12
path	-1	2	0	5	1	2

# Dijkstra算法实现

- 至此，0 到其他各顶点的最短路径长度都求完了。
- 想要追踪  $v_0$  到  $v_k$  的最短路径，只需要从  $\text{path}[v_k]$  开始倒向追踪即可。



	0	1	2	3	4	5
S	1	1	1	<u>1</u>	0	1
dist	0	20	5	22	28	12
path	-1	2	0	5	1	2

下节课再见