

# Prim算法实现



主讲人：邓哲也



# 大纲

Prim算法实现

Prim算法执行过程

# Prim算法实现

假设采用邻接矩阵来存储图。

在 Prim 算法运算过程中，需要知道以下两类信息：

集合  $T'$  内各顶点距离  $T$  内各顶点权值最小的边的权值

集合  $T'$  内各顶点距离  $T$  内哪个顶点最近

# Prim算法实现

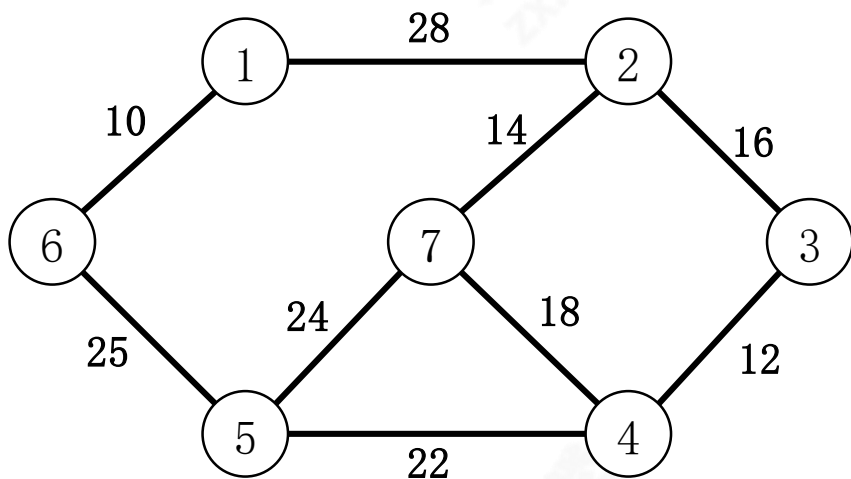
为了存储这两类信息，我们使用两个辅助数组：

$\text{low}[x]$ ：存放顶点集合  $T'$  内的顶点  $x$  到顶点集合  $T$  内各顶点权值最小的边的权值。

$\text{pos}[x]$ ：存放顶点集合  $T'$  内的顶点  $x$  到顶点集合  $T$  内那个顶点最近。当  $\text{pos}[x] = 0$  时，表示  $x$  已经放入  $T$  中。

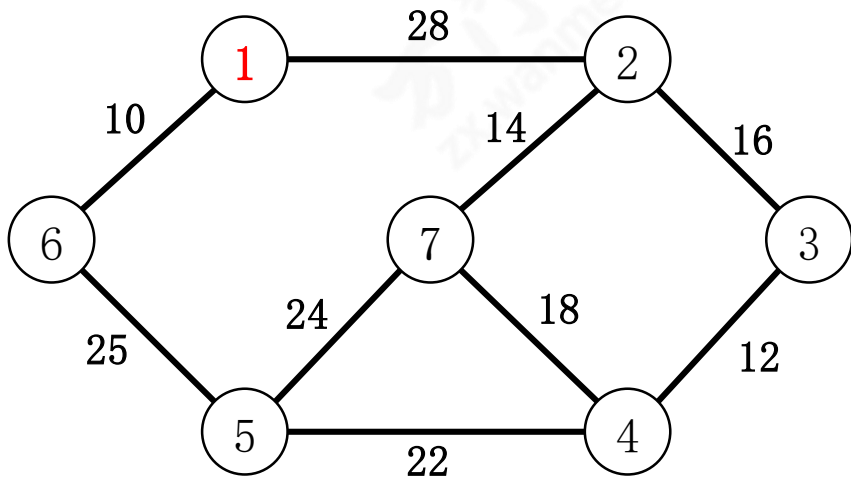
# Prim算法执行过程

以下面这个无向图为例，我们来看看 Prim 算法执行过程。



# Prim算法执行过程

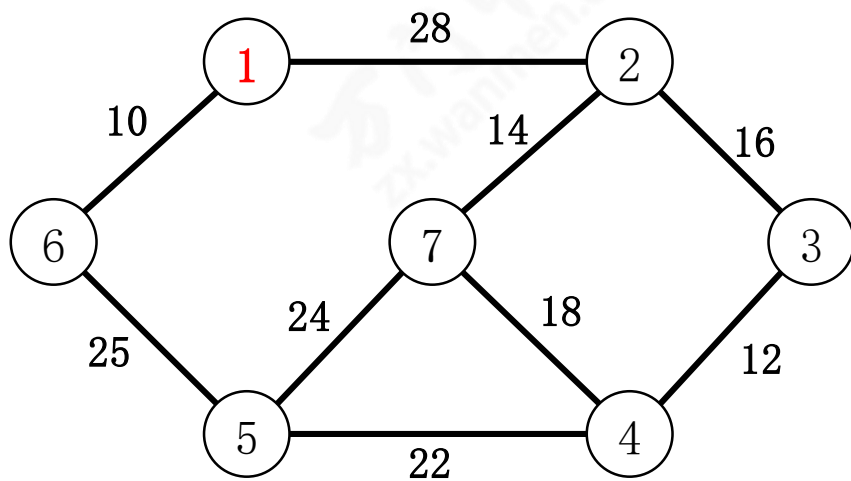
假设选择的起始顶点是1，则两个辅助数组的初始状态如下表：



	1	2	3	4	5	6	7
low	$\emptyset$	28	$\infty$	$\infty$	$\infty$	10	$\infty$
pos	$\emptyset$	1	1	1	1	1	1

# Prim算法执行过程

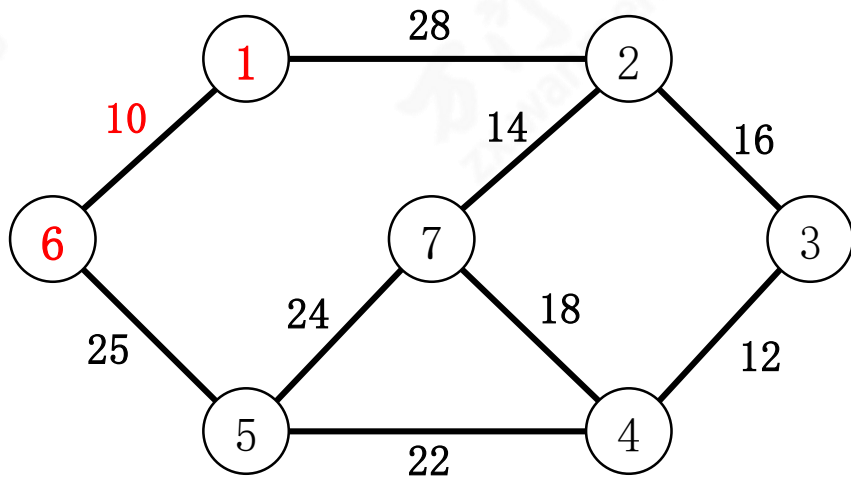
low[6]最小, 选择 6 号点加入 T



	1	2	3	4	5	6	7
low	0	28	$\infty$	$\infty$	$\infty$	10	$\infty$
pos	0	1	1	1	1	1	1

# Prim算法执行过程

low[5]最小, 选择 5 号点加入 T:

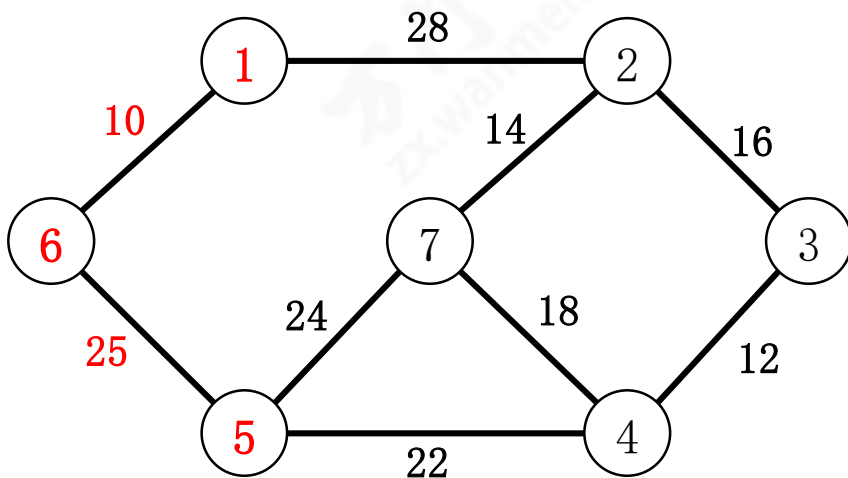


	1	2	3	4	5	6	7
low	0	28	$\infty$	$\infty$	25	10	$\infty$
pos	0	1	1	1	6	0	1



# Prim算法执行过程

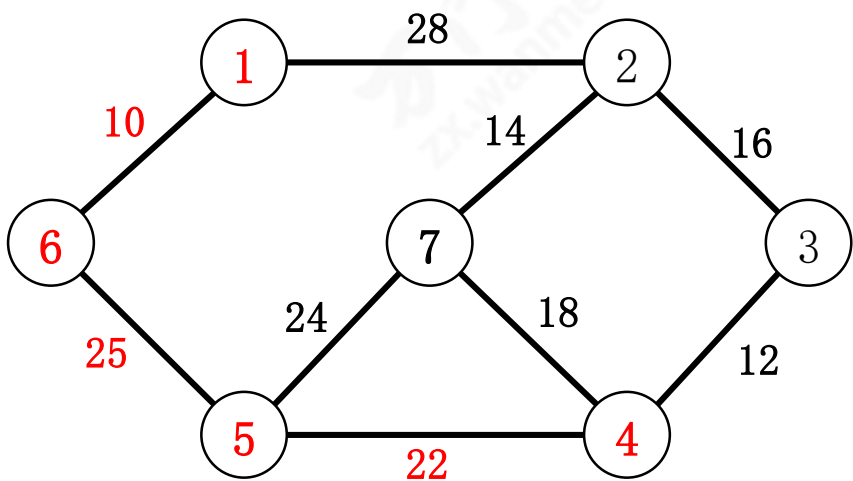
low[4] 最小, 选择 4 号点加入 T:



	1	2	3	4	5	6	7
low	0	28	$\infty$	22	25	10	24
pos	0	1	1	5	0	0	5

# Prim算法执行过程

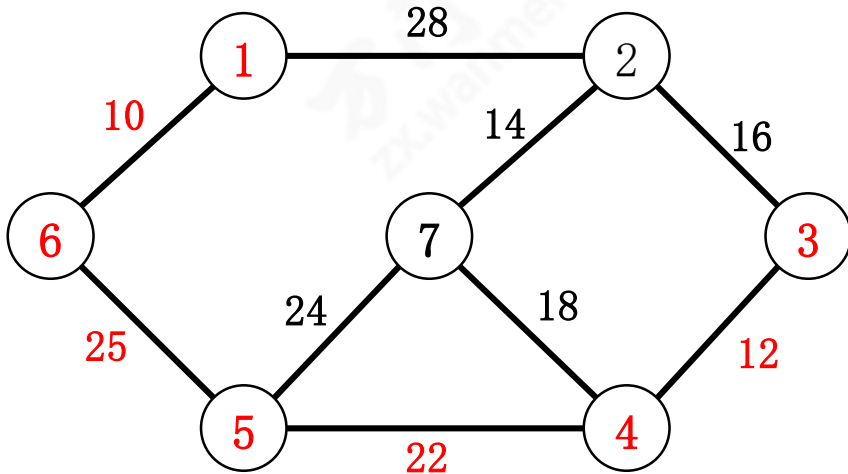
low[3] 最小，选择 3 号点加入 T:



	1	2	3	4	5	6	7
low	0	28	12	22	25	10	18
pos	0	1	4	0	0	0	4

# Prim算法执行过程

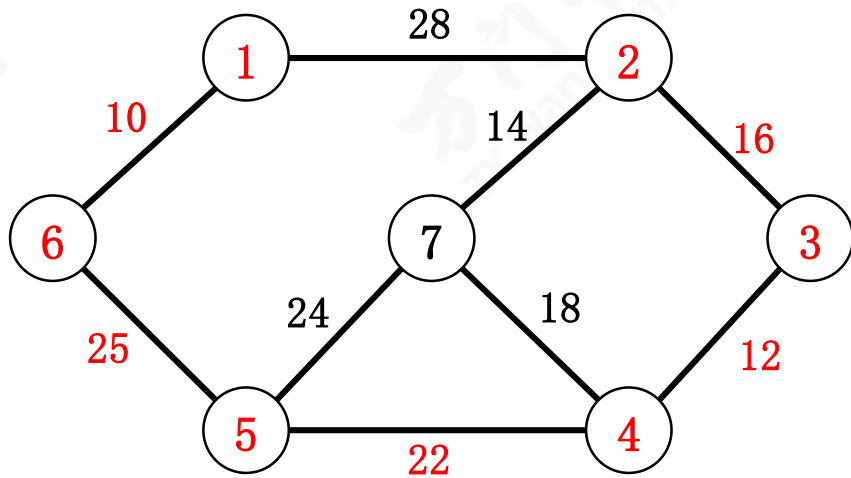
low[2] 最小, 选择 2 号点加入 T:



	1	2	3	4	5	6	7
low	0	16	12	22	25	10	18
pos	0	3	0	0	0	0	4

# Prim算法执行过程

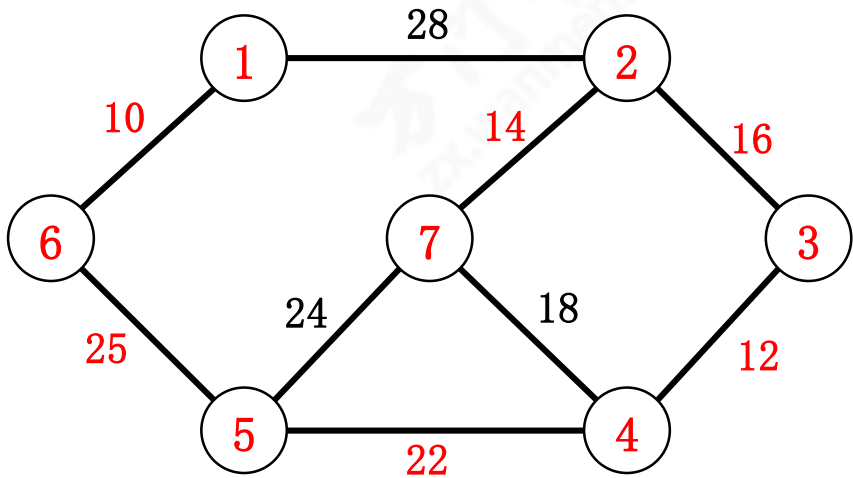
low[7] 最小, 选择 7 号点加入 T:



	1	2	3	4	5	6	7
low	∅	16	12	22	25	10	14
pos	∅	∅	∅	∅	∅	∅	2

# Prim算法执行过程

算法结束。



	1	2	3	4	5	6	7
low	0	16	12	22	25	10	14
pos	0	0	0	0	0	0	0

下节课再见