本节主题：

最小生成树的普里姆算法

普里姆(Prim)算法

- 问题
  - G=(V，E)是一个具有n个顶点的带权连通无向图
  - T=(U，TE)是G的最小生成树，其中U是T的顶点集，TE是T的边集
- 由G构造最小生成树T的步骤

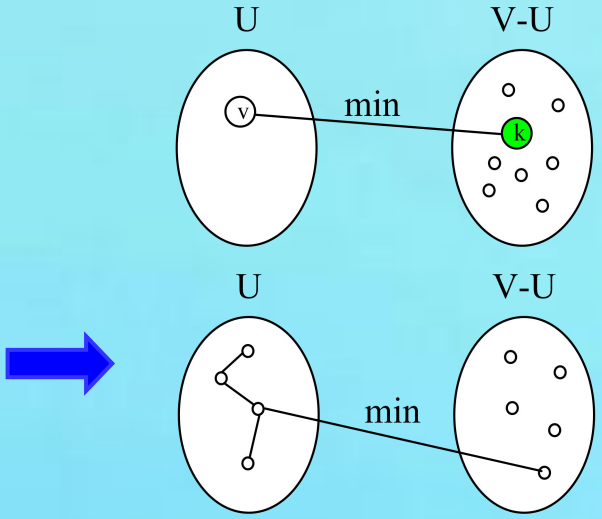
（1）初始化U={v}，v到其他顶点的所有边为候选边；

（2）重复以下步骤n-1次，在U中加入其他n-1个顶点

① 从候选边中挑选权值最小的边加入TE，设该边在V-U中的顶点是k，将k加入U中；

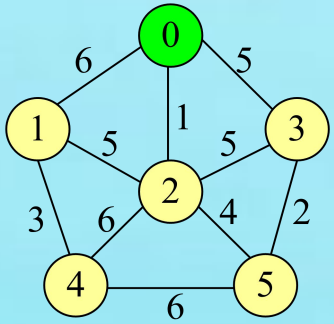
② 考察当前V-U中的所有顶点i，修改候选边：若(k,i)的权值小于原来和顶点k关联的候选边，则用(k,i)取代后者作为候选边。

Robert Clay Prim
1957年提出



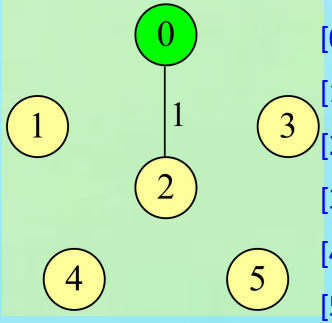
普里姆算法过程演示

```
int lowcost[MAXV]; //记录从U到U-V的边的最小权值
int closest[MAXV]; //记录最小权值的边对应的顶点
```



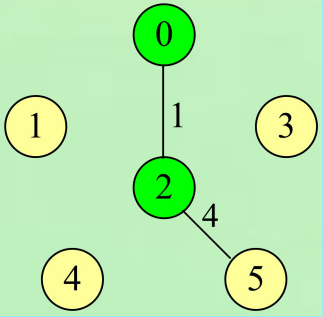
	lowcost	closest
[0]	0	0
[1]	6	0
[2]	1	0
[3]	5	0
[4]	∞	0
[5]	∞	0

U={0}



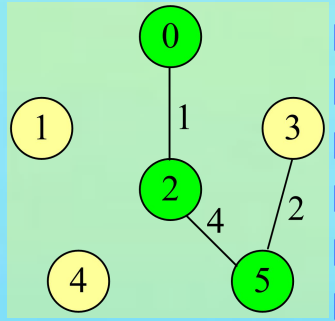
	lowcost	closest
[0]	0	0
[1]	5	2
[2]	0	0
[3]	5	0
[4]	6	2
[5]	4	2

U={0,2}



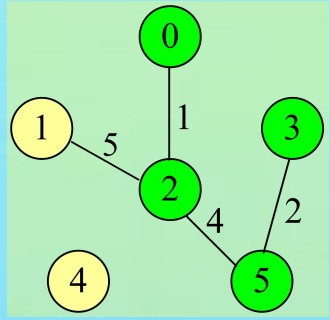
	lowcost	closest
[0]	0	0
[1]	5	2
[2]	0	0
[3]	2	5
[4]	6	2
[5]	0	2

U={0,2,5}



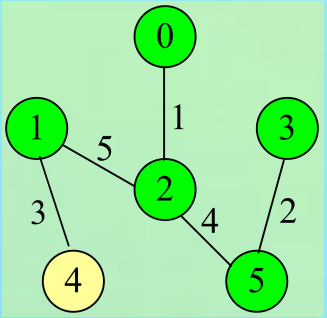
	lowcost	closest
[0]	0	0
[1]	5	2
[2]	0	0
[3]	0	5
[4]	6	2
[5]	0	2

U={0,2,5,3}



	lowcost	closest
[0]	0	0
[1]	0	2
[2]	0	0
[3]	0	5
[4]	3	1
[5]	0	2

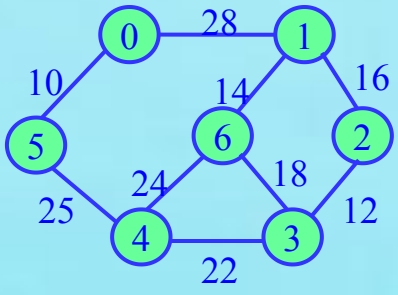
U={0,2,5,3,1}



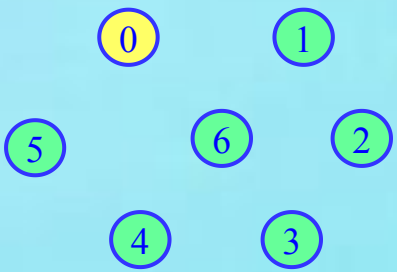
	lowcost	closest
[0]	0	0
[1]	0	2
[2]	0	0
[3]	0	5
[4]	0	1
[5]	0	2

U={0,2,5,3,1,4}

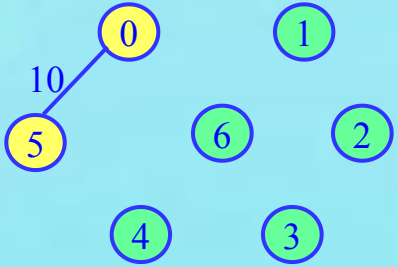
再例：普里姆算法过程



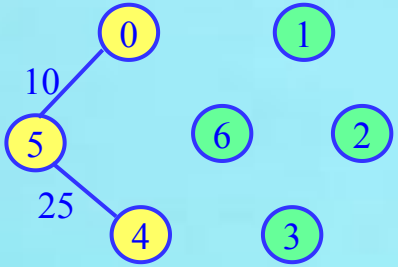
图G



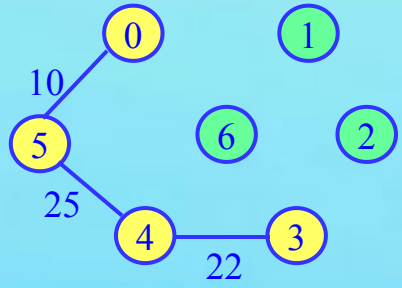
U={0}



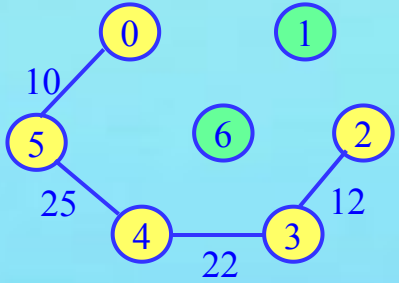
U={0,5}



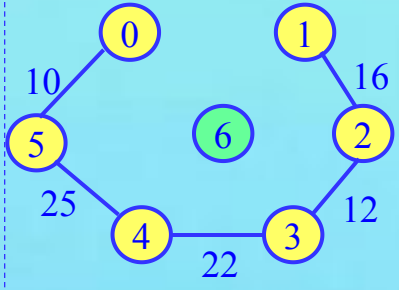
U={0,5,4}



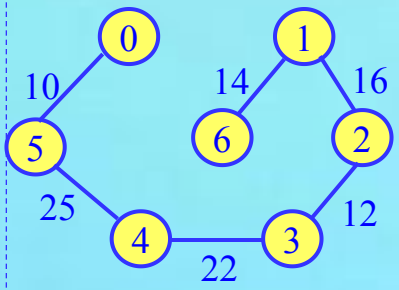
U={0,5,4,3}



U={0,5,4,3,2}



U={0,5,4,3,2,1}



U={0,5,4,3,2,1,6}

```
void Prim(MGraph g,int v)
{
  //变量声明及初始化
  for (i=1; i<g.n; i++)
  {
    //找权值最小的邻接点k
    min=INF;
    for (j=0; j<g.n; j++)
      if (lowcost[j]!=0 && lowcost[j]<min)
      {
        min=lowcost[j];
        k=j;
      }
    printf(" 边(%d,%d): %d\n",closest[k],k,min);
    //调整lowcost和closest
    lowcost[k]=0;
    for (j=0; j<g.n; j++)
      if (g.edges[k][j]!=0 && g.edges[k][j]<lowcost[j])
      {
        lowcost[j]=g.edges[k][j];
        closest[j]=k;
      }
  }
}
```

```
int lowcost[MAXV];
int min;
int closest[MAXV],i,j,k;
for (i=0; i<g.n; i++)
{
  lowcost[i]=g.edges[v][i];
  closest[i]=v;
}
```

$$\begin{bmatrix} 0 & 6 & 1 & 5 & \infty & \infty \\ 6 & 0 & 5 & \infty & 3 & \infty \\ 1 & 5 & 0 & 5 & 6 & 4 \\ 5 & \infty & 5 & 0 & \infty & 2 \\ \infty & 3 & 6 & \infty & 0 & 6 \\ \infty & \infty & 4 & 2 & 6 & 0 \end{bmatrix}$$

	lowcost	closest
[0]	0	0
[1]	6	0
[2]	1	0
[3]	5	0
[4]	∞	0
[5]	∞	0

	lowcost	closest
[0]	0	0
[1]	5	2
[2]	0	0
[3]	5	0
[4]	6	2
[5]	4	2

	lowcost	closest
[0]	0	0
[1]	5	2
[2]	0	0
[3]	2	5
[4]	6	2
[5]	0	2

. . . . . .