

{% note info %} **摘要** Title: 858. Prim算法求最小生成树 Tag: Prim、最小生成树 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

[@TOC](#)

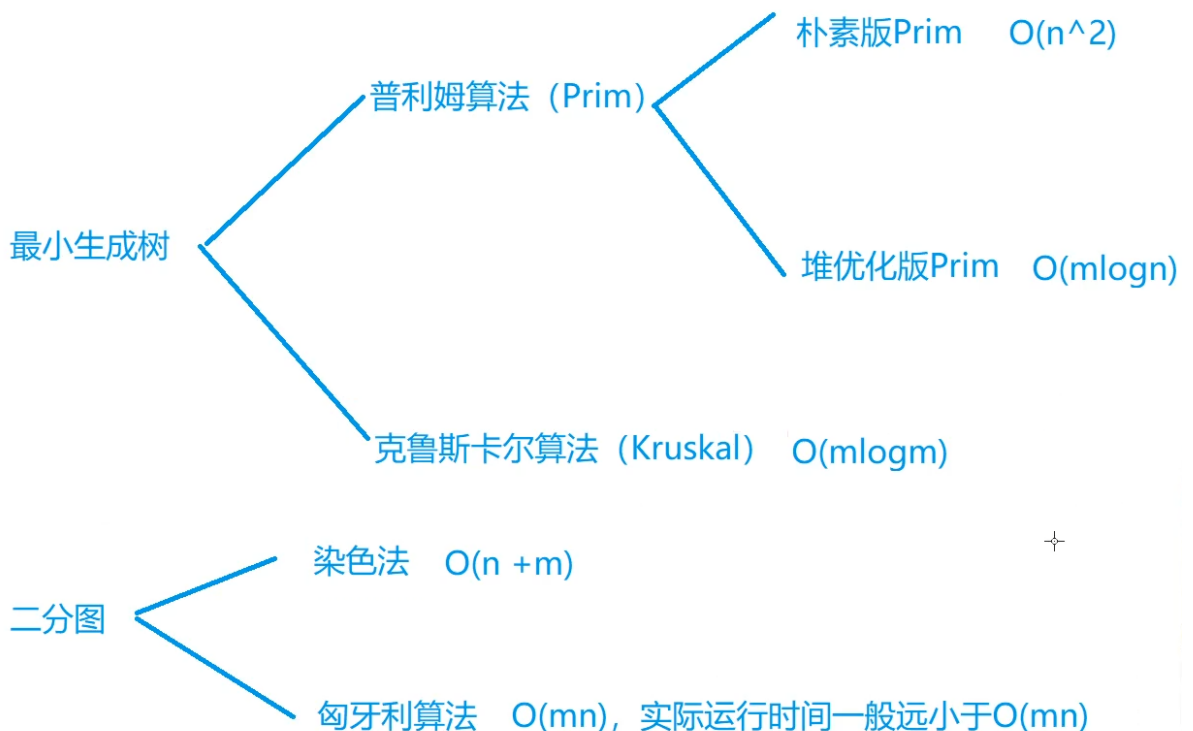
## 858. Prim算法求最小生成树

- 题意

给定一个  $n$  个点  $m$  条边的无向图，图中可能存在重边和自环，边权可能为负数。求最小生成树的树边权重之和，如果最小生成树不存在则输出 impossible。给定一张边带权的无向图  $G=(V,E)$ ，其中  $V$  表示图中点的集合， $E$  表示图中边的集合， $n=|V|$ ， $m=|E|$ 。由  $V$  中的全部  $n$  个顶点和  $E$  中  $n-1$  条边构成的无向连通子图被称为  $G$  的一棵生成树，其中边的权值之和最小的生成树被称为无向图  $G$  的最小生成树。

- 思路

### 最小生成树和二分图汇总



### 注意

- 稠密图一般用 朴素版的Prim
- 稀疏图一般用 Kruskal

## 朴素版的Prim

① 朴素 Prim

 $dist[i] \leftarrow +\infty$ 

57164

for ( $i=0; i < n; i++$ )

找到集合外距离最近的点

用t更新其他点到集合的距离

 $dist[t] = true;$ 

{% note info %} 与朴素版的Dijkstra区别 区别1

- Dijkstra的t, 是不在最短路集合中的离起点最近的点
- Prim的t, 是集合外最近的点

区别2

- Dijkstra是用t来更新其他点到起点的距离
- Prim是用t来更新其他点到集合的距离

区别3

- Dijkstra外层迭代n - 1次
- Prim外层迭代n次 {% endnote %}

注意:

- 集合是指当前已经在连通块中的点
- u到集合的距离: u到集合内的点的最短距离

## • 代码

...

Author: NEFU AB-IN

Date: 2022-03-03 18:52:55

FilePath: \ACM\Acwing\858.py

```

LastEditTime: 2022-03-03 19:13:17
...
N = 550
INF = int(2e9)
g = [[INF] * N for _ in range(N)]
st, dist = [0] * N, [INF] * N

def prim():
    res = 0
    for i in range(n): #迭代n次
        t = -1
        for j in range(1, n + 1): #选出最小的
            if (st[j] == 0 and (t == -1 or dist[t] > dist[j])):
                t = j
        if i and dist[t] == INF: #如果不是第一个点, 并且最小的距离为INF, 那么不
存在
            return INF
        if i:
            res += dist[t] #如果不是第一个点, 那么加上距离
        for j in range(1, n + 1):
            dist[j] = min(dist[j], g[j][t]) #用t更新距离, 是对集合的, 所以不加
dist[t]
            st[t] = 1
    return res

n, m = map(int, input().split())
for i in range(m):
    x, y, z = map(int, input().split())
    g[x][y] = g[y][x] = min(g[x][y], z) #双向边
res = prim()
if res == INF:
    print("impossible")
else:
    print(res)

```