图论

最小生成树

- 最小生成树算法
  - Kruskal
  - Prim
- 算法的正确性证明
- 次小生成树

最小生成树

- 最小生成树算法
  - Kruskal
  - Prim
- 算法的正确性证明

EXCLUSIVE-OR

- 有n个数X0~Xn-1，开始时你不知道它们的值，逐步给你信息：1、直接给出 Xp 的值v。2、给出Xp ^ Xq的值v。
- 然后在线询问某些(<=15个) Xi 异或的值。
- N <= 20000 , Q <= 40000

THE UNIQUE MST

- 给定一个无向图，判断最小生成树是否唯一
- N <= 5000
- 扩展： 求次小生成树

游览计划

- 给定N*M的长方形，选最少权值和的格子使得要求的K个点连通。 N,M <= 10
- 第一行有两个整数，N和 M，描述方块的数目。 接下来 N 行， 每行有 M 个非负整数， 如果该整数为 0， 则该方块为一个景点；否则表示控制该方块至少连接的价格。

1	4	1	3	4	2	4	1
4	3	1	2	沈园	1	2	3
3	2	1	3	八字桥	3	1	2
2	6	5	周恩来故居	2	4	1	东湖
5	1	2	1	3	4	2	5
5	1	3	1	5	大禹陵	1	4
5	兰亭	6	1	4	5	3	4
鉴湖	2	2	2	3	4	1	1

斯坦纳树

- 包含给定顶点的最小生成树

- 状态压缩，把k个节点的连通状态用一个二进制数j表示
- dp[i][j]表示以i为根，与对应状态为j的节点连通的子树的最小权值
- 有两种转移方法：
  - 枚举子树的形态：dp[i][j]=min(dp[i][j]，dp[i][k]+dp[i][l])，其中kk和ll是对j的一个划分
  - 按照边进行松弛：dp[i][j]=min(dp[i][j]，dp[i′][j]+w[i][i′])，其中ii和i′i′之间有边相连

斯坦纳树

- 包含给定顶点的最小生成树

- 状 … 制数j表示
- d … 通的子树的最小权值
- 有
  - … dp[i][k]+dp[i][l])，其中kk和ll是对j
  - … dp[i′][j]+w[i][i′])，其中ii和i′i′之间

```
枚举状态集S
{
        枚举S的子集s
        {
              更新f[S][1~n]
        }
        将 f[S][x]<inf 的x入队
        spfa(S)
}
```

管道管理

- 给定n点m边的图，连接边(u,v)需要花费w，问满足使k个点中同颜色的点都连通的最小费用。
- N <= 1000 , M <= 3000

最短路径

- 最短路径算法
  - Dijkstra （堆优化）
  - SPFA
  - Floyd 【传递闭包】
- 次短路径

ROADBLOCKS

- N个点，M条无向边，找出从1到N的次短路径。次短路径的定义是比不同于最短路径的一条路径。这里每条边可以走多次。

- N <= 5000 ; M <= 100000

INVADE THE MARS

- 起点是1, 终点是n , 某些城市会受到其他城市的保护, 受到保护的城市必须在保护该城市的城市被攻陷后才能被进入, 还有一点是军队是同时出发的, 也就是军队可以在受保护的城市外面停留, 等到保护其城市被攻陷后军队可以马上进入该座城市, 也就是说进入一座城市的时间是max(第一次到达， 保护其城市的城市被攻陷的时间). 问到达n点的最少时间是多少.
- N <= 3000 M <= 70000

差分约束系统

- 差分约束系统，就是给出一组形如 xi - xj <= d的不等式，求出这组不等式的一组解。这类问题通常转化为图论中的最短路来解。
  - a - b >(=) c
  - a - b <(=) c
  - a - b = c
  - a >/=/< c

差分约束系统

- 差分约束系统，就是给出一组形如 $x_i - x_j <= d$的不等式，求出这组不等式的一组解。这类问题通常转化为图论中的最短路来解。
  - 求解最大值
  - 求解最小值
  - 求解是否可行

LAYOUT

- 有一个长度为N的数组a[1..n]，每次告诉你ai到aj之和大于或者小于某一个值，问这个数组是否存在。
- N <= 100000

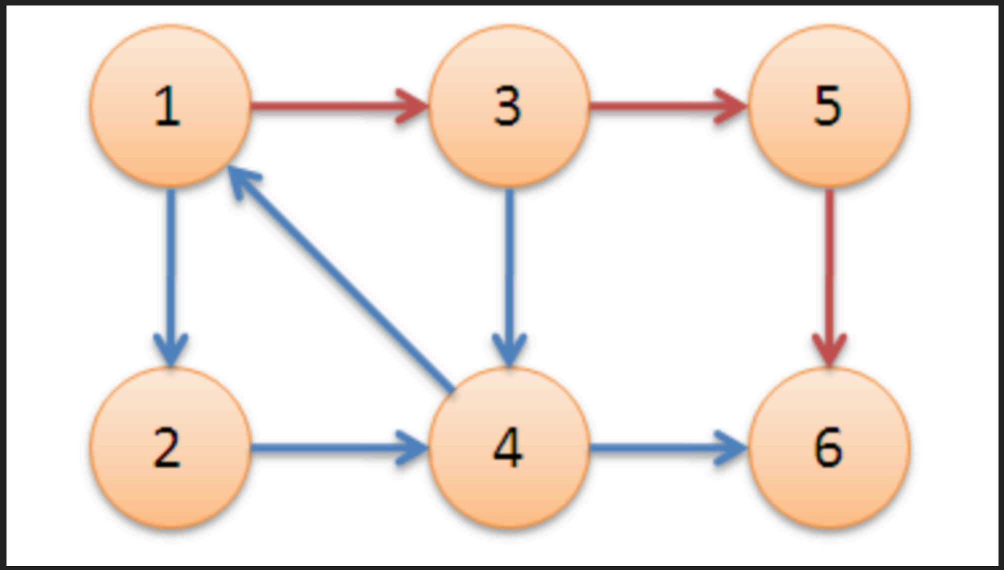
环

- 判断图是否有环
  - 拓扑排序
- 找出图中的环
  - 求无向图的最大简单环
- 强连通分量
- 图的割点、割边

TARJAN算法

- DFN(u)为节点u搜索的次序编号(时间戳)
- Low(u)为u或u的子树能够追溯到的最早的栈中节点的次序号 tarjan(u)
-

```
tarjan(u)
{
    DFN[u]=Low[u]=++Index
    Stack.push(u)
    for each (u, v) in E
        if (v is not visted)
            tarjan(v)
            Low[u] = min(Low[u], Low[v])
        else if (v in S)
            Low[u] = min(Low[u], DFN[v])
    if ( 判断 ) { ... }
}
```



TARJAN算法

- 强连通分量
  - DFN[u] == LOW[u]
  - u 栈内所有元素就是一个强连通分量
- 割点 u
  - LOW[v] >= DFN[u]   (u, v) in E
- 割边 v-u
  - LOW[v] > DFN[u]

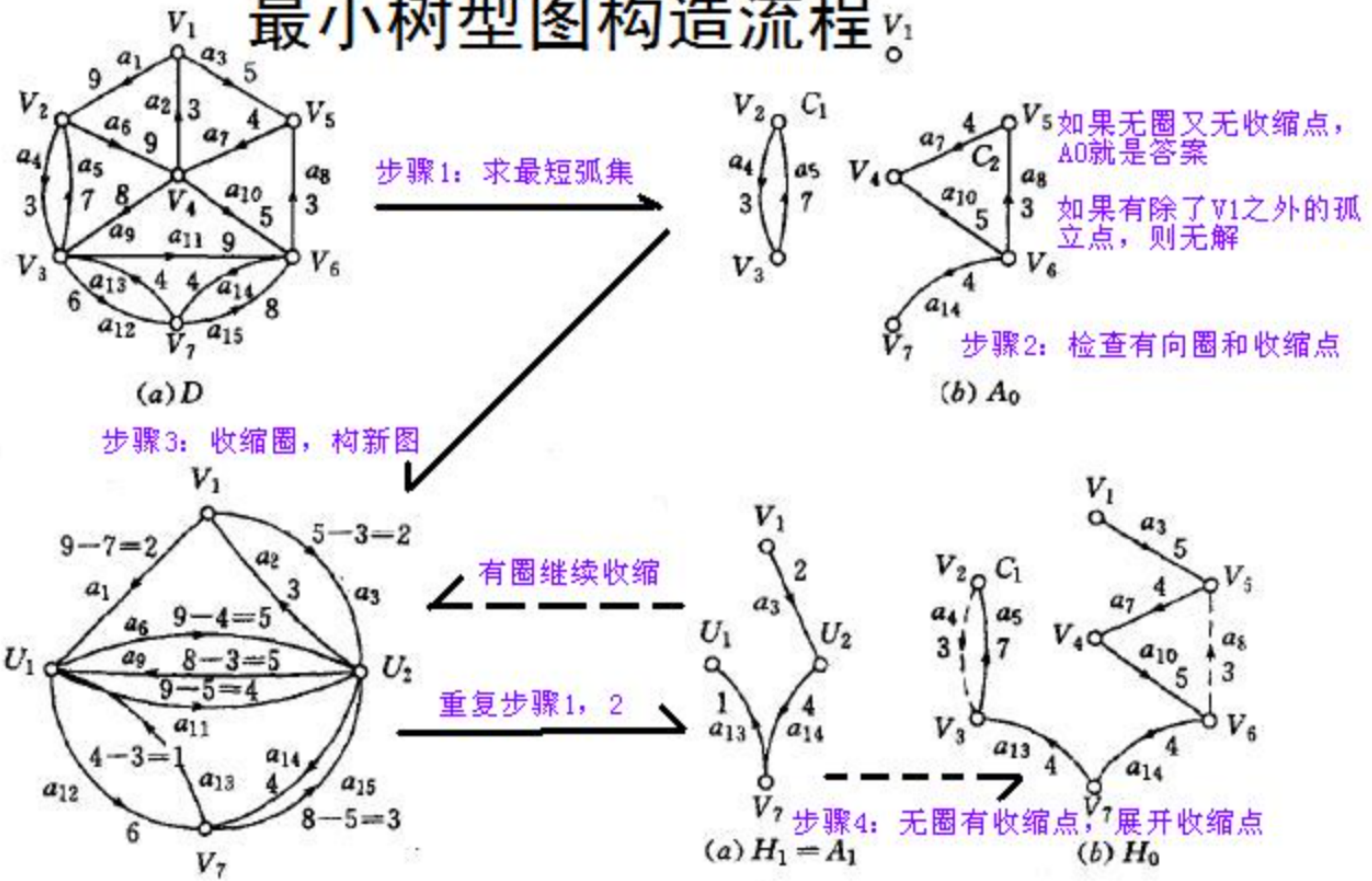
最小树形图

- 设G = (V, E)是一个有向图,如果具有下述性质
- (1)G中不包含有向环;
- (2)存在一个顶点vi,它不是任何弧的终点,而V的其它顶点都恰好是唯一的一条弧的终点.则称 G是以vi为根的树形图.
- 最小树形图就是有向图G = (V, E)中以vi为根的树形图中权值的和最小的那一个.

最小树形图

- 找到除了root以为其他点的权值最小的入边。用mincost[i]记录; 如果出现除了root以为存在其他孤立的点，则不存在最小树形图
- 找到图中所有的环，并对环进行缩点，重新编号,更新其他点到环上的点的距离：
  - g[v][Vk]=min { g[v][Vkj]-mincost[Vkj] }
  - g[Vk][v]=min { g[Vkj][v] }

最小树型图构造流程


步骤1：求最短弧集
如果无圈又无收缩点，A0就是答案
如果有除了V1之外的孤立点，则无解
步骤2：检查有向圈和收缩点
(a) D
(b) A_0
步骤3：收缩圈，构新图
有圈继续收缩
重复步骤1，2
步骤4：无圈有收缩点，展开收缩点
(a) $H_1 = A_1$
(b) H_0

二分图匹配