

Naticl's blog

树的最小支配集 最小点覆盖 与 最大独立集 (图论)

首先看一下三者的定义：

**定义1** 对于图 $G=(V,E)$ 来说，**最小支配集**指的是从 $V$ 中取尽量少的点组成一个集合，使得对于 $V$ 中剩余的点都与取出来的点有边相连。也就是说，设 $V'$ 是图 $G$ 的一个支配集，则对于图中的任意一个顶点 $u$ ，要么属于集合 $V'$ ，要么与 $V'$ 中的顶点相邻。在 $V'$ 中出去任何元素后 $V'$ 不再是支配集，则支配集是极小支配集。称 $G$ 的所有支配集中顶点个数最少的支配集为最小支配集，最小支配集中顶点的个数称为支配数。

**定义2** 对于图 $G=(V,E)$ 来说，**最小点覆盖**指的是从 $V$ 中取尽量少的点组成一个集合，使得 $E$ 中所有的边都与取出来的点相连。也就是说，设 $V'$ 是图 $G$ 的一个顶点覆盖，则对于图中的任意一条边 $(u,v)$ ，要么 $u$ 属于集合 $V'$ ，要么 $v$ 属于集合 $V'$ 。在 $V'$ 中除去任何元素后 $V'$ 不在是顶点覆盖，则 $V'$ 是极小顶点覆盖。称 $G$ 的所有顶点覆盖中顶点个数最少的覆盖为最小点覆盖。

**定义3** 对于图 $G=(V,E)$ 来说，**最大独立集**指的是从 $V$ 中取尽量多的点组成一个集合，使得这些点之间没有边相连。也就是说，设 $V'$ 是图 $G$ 的一个独立集，则对于图中任意一条边 $(u,v)$ ， $u$ 和 $v$ 不能同时属于集合 $V'$ ，甚至可以 $u$ 和 $v$ 都不属于集合 $V'$ 。在 $V'$ 中添加任何不属于 $V'$ 元素后 $V'$ 不再是独立集，则 $V'$ 是极大独立集。称 $G$ 的所有顶点独立集中顶点个数最多的独立集为最大独立集。

对于任意图 $G$ 来说，这三个问题不存在多项式时间的解法。不过对于树来说，却很容易。目前有两种解法，一种基于贪心思想，另一种基于树形动态规划思想。

一：贪心法

基本算法：

以最小支配集为例，对于树上的最小支配集问题，贪心策略是首先选择一点为根，按照深度优先遍历得到遍历序列，按照所得序列的反向序列的顺序进行贪心，对于一个既不属于支配集也不与支配集中的点相连的点来说，如果他的父节点不属于支配集，将其父节点加入支配集。

这里注意到贪心的策略中贪心的顺序非常重要，按照深度优先遍历得到遍历序列的反向进行贪心，可以保证对于每个点来说，当期子树都被处理过后才轮到该节点的处理，保证了贪心的正确性。

- ①以1号点深度优先搜索整棵树，求出每个点在深度优先遍历序列中的编号和每个点的父节点编号。
- ②按照深度优先序列的反向序列检查，如果当前点既不属于支配集也不与支配集中的点相连，且他的父节点不属于支配集，将其父节点加入支配集，支配集中点的个数加1.标记当前节点、当前节点的父节点和当前节点的父节点的父节点，因为这些节点要么属于支配集，要么与支配集中的点相连。

最小点覆盖于最大独立集与上面的做法相似。对于最小点覆盖来说，贪心的策略是，如果当前点和当前点的父节点都不属于顶点覆盖集合，则将父节点加入到顶点覆盖集合，并标记当前节点和其父节点都被覆盖。对于最大独立集来说，贪心策略是如果当前点没有被覆盖，则将当前节点加入到独立集，并标记当前节点和其父节点都被覆盖。

需要注意的是由于默认程序中根节点和其他节点的区别在于根节点的父节点是其自己，所以三种问题对根节点的处理不同。对于最小支配集和最大独立集，需要检查根节点是否满足贪心条件，但是对于最小点覆盖不可以检查根节点。

具体实现：

采用链式前向星存储整棵树。对于DFS(), newpos[i]表示深度优先遍历序列的第i个点是哪个点，now表示当前深度优先遍历序列已经有多少个点了。select[]用于深度优先遍历序列的判重，p[i]表示点i的父节点的编号。对于greedy(), s[i]如果为true，表示第i个点被覆盖。set[i]表示点i属于要求的点集。

首先是深度优先遍历，得到遍历序列。代码如下：

```
int p[maxn];
bool select[maxn];
```

公告

昵称： Kevin\_naticl  
园龄： 3年10个月  
粉丝： 16  
关注： 21  
+加关注

2021年9月						
日	一	二	三	四	五	六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

搜索

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

随笔分类

动态规划dp(58)  
二分答案(7)  
模拟(2)  
扫描线(1)  
数据结构- K-D Tree(1)  
数据结构- Treap / Splay(2)  
数据结构- Trie 树(11)  
数据结构- 其他可持久化结构(2)  
数据结构-并查集(4)  
数据结构-堆/单调队列(9)  
数据结构-树链剖分(10)  
数据结构-线段树(10)  
数据结构-栈(5)  
数据结构-主席树(5)  
数据结构-左偏树(1)  
更多

随笔档案

2019年11月(2)  
2019年7月(9)  
2018年11月(2)  
2018年10月(15)

```
int newpos[maxn];
int now;
int n,m;
void DFS(int x)
{
    newpos[now++]=x;
    int k;
    for(k=head[x];k!=-1;k=edge[k].next)
    {
        if(!select[edge[k].to])
        {
            select[edge[k].to]=true;
            p[edge[k].to]=x;
            DFS(edge[k].to);
        }
    }
}
```



对于最小支配集，贪心函数如下：



```
int greedy()
{
    bool s[maxn];
    bool set[maxn]={0};
    int ans=0;
    int i;
    for(i=n-1;i>=0;i--)
    {
        int t=newpos[i];
        if(!s[t])
        {
            if(!set[p[t]])
            {
                set[p[t]]=true;
                ans++;
            }
            s[t]=true;
            s[p[t]]=true;
            s[p[p[t]]]=true;
        }
    }
    return ans;
}
```



对于最小点覆盖，贪心函数如下：



```
int greedy()
{
    bool s[maxn]={0};
    bool set[maxn]={0};
    int ans=0;
    int i;
```

- 2018年9月(29)
- 2018年8月(20)
- 2018年7月(15)
- 2018年6月(25)
- 2018年5月(30)
- 2018年4月(17)
- 2018年3月(2)
- 2018年1月(3)
- 2017年12月(5)
- 2017年11月(1)

相册

KKK(4)

阅读排行榜

- 1. 树的最小支配集 最小点覆盖 与 最大独立集 (图论)(1535)
- 2. AC自动机详解 (P3808 模板)(1268)
- 3. (C++一本通)最少转弯问题 (经典搜索)(1152)
- 4. 主席树模板(761)
- 5. [BZOJ4260] Codechef REBXOR (01字典树,异或前缀和)(747)

评论排行榜

- 1. 树的最小支配集 最小点覆盖 与 最大独立集 (图论)(4)
- 2. [USACO Section 4.4]追查坏牛奶Pollutant Control (最小割)(3)
- 3. P1993 小K的农场 (差分约束)(3)
- 4. AC自动机详解 (P3808 模板)(2)
- 5. [BZOJ3261] 最大异或和 (异或前缀和,可持久化Trie)(1)

推荐排行榜

- 1. AC自动机详解 (P3808 模板)(3)
- 2. 树的最小支配集 最小点覆盖 与 最大独立集 (图论)(2)
- 3. P2634 [国家集训队]聪聪可可(2)
- 4. 点分治模板(2)
- 5. 主席树模板(2)

最新评论

- 1. Re:[USACO Section 4.4]追查坏牛奶 Pollutant Control (最小割)  
@ Kevin\_naticl噢好的，没关系，非常感谢你啦！ qwq...  
--顾玥\_浅笑
- 2. Re:[USACO Section 4.4]追查坏牛奶 Pollutant Control (最小割)  
@ 顾玥\_浅笑因为处理反向边的时候会有问题。cost[i]-=w;cost[i^1]+=w;used+=w;而1^1 是 0 然而 0 这条边我们是访问不到的。年代久远,回复的有点晚哈 QwQ....  
--Kevin\_naticl
- 3. Re:[USACO Section 4.4]追查坏牛奶 Pollutant Control (最小割)  
你好！请问tot为什么要从1开始呢？我在洛谷上交题，tot=0就WA了，调了很久...QAQ  
--顾玥\_浅笑
- 4. Re:树的最小支配集 最小点覆盖 与 最大独立集 (图论)  
@ Kevin\_naticl好像那个是!s[t]&&!s[p[t]]才对吧...

```

for(i=n-1;i>=1;i--)
{
    int t=newpos[i];
    if(!s[t]&& s[p[t]])
    {
        set[p[t]]=true;
        ans++;
        s[t]=true;
        s[p[t]]=true;
    }
}
return ans;
}

```

对于最大独立集，贪心函数如下：

```

int greedy()
{
    bool s[maxn]={0};
    bool set[maxn]={0};
    int ans=0;
    int i;
    for(i=n-1;i>=0;i--)
    {
        int t=newpos[i];
        if(!s[t])
        {
            set[t]=true;
            ans++;
            s[t]=true;
            s[p[t]]=true;
        }
    }
    return ans;
}

```

使用样例：

```

int main()
{
    /*读入图信息*/
    memset(select,0,sizeof(select));
    now=0;
    select[1]=true;
    p[1]=1;
    DFS(1);
    printf("%d\n",greedy());
}

```

该方法经过一次深度优先遍历和一次贪心得到最终解，第一步的时间复杂度为 $O(m)$ ，由于这是一棵树， $m=n-1$ 。第二步是 $O(n)$ ，总的是 $O(n)$ 。

## 二.树形动态规划法