

动态规划 3

树形背包问题

基本模型

给定一棵 n 阶树，每个结点拥有重量 w_i 和价值 v_i 且选取某个结点必须选取该结点的父结点。

给定背包容量 M 问可以得到的最大价值。

dfs 序优化

考虑后序遍历结点并编号，设 $\text{dp}(i, j)$ 表示仅考虑后序遍历前 i 个结点的容量为 j 的背包的最大价值。

考虑新加入的第 i 个结点，发现后序遍历前 $[i - \text{sz}_i + 1, i]$ 个结点代表结点 i 的子树。

分别考虑不选取第 i 个结点和选取第 i 个结点的情况，于是有状态转移方程

$$\text{dp}(i, j) = \max(\text{dp}(i - \text{sz}_i, j), \text{dp}(i - 1, j - w_i) + v_i)$$

时间复杂度 $O(nm)$

```
void dfs(int u){
    sz[u]=1;
    for(int i=head[u]; i; i=edge[i].next){
        int v=edge[i].to;
        dfs(v);
        sz[u]+=sz[v];
    }
    dfs_t++;
    _for(i, 0, w[u])
        dp[dfs_t][i]=dp[dfs_t-sz[u]][i];
    _for(i, w[u], MAXM)
        dp[dfs_t][i]=max(dp[dfs_t-1][i-w[u]]+v[u], dp[dfs_t-sz[u]][i]);
}
```

点对优化

题意

[洛谷p1273](#)

给定一棵以 1 为根的 n 阶有根树，树的每条边有费用 f_i 树的叶节点有价值 v_i

选取若干条边的利润为所有可以通过选取边与根节点连通的叶节点的价值和减去所有选取边的费用和。

求利润非负时最多可以选取的边数。

题解

设 $dp(i,j)$ 表示以 i 为根节点的子树中选取 j 个叶子节点的最大盈利，不难得出状态转移方程。

考虑以类似枚举 LCA 的方式合并子节点结果，可以 $O(n^2)$ 完成所有状态转移。最后查询满足 $dp(1,i) \geq 0$ 的最大的 i 即可。

```
const int MAXN=3e3+5,Inf=1e9;
LL dp[MAXN][MAXN];
int head[MAXN],edge_cnt;
struct Edge{
    int to,w,next;
}edge[MAXN];
void Insert(int u,int v,int w){
    edge[++edge_cnt]=Edge{v,w,head[u]};
    head[u]=edge_cnt;
}
int sz[MAXN];
void dfs(int u){
    for(int i=head[u];i;i=edge[i].next){
        int v=edge[i].to;
        dfs(v);
        for(int j=sz[u];j>=0;j--)_rep(k,1,sz[v])
            dp[u][j+k]=max(dp[u][j+k],dp[u][j]+dp[v][k]-edge[i].w);
        sz[u]+=sz[v];
    }
}
int main()
{
    int n=read_int(),m=read_int();
    _rep(i,1,n)_rep(j,1,n)dp[i][j]=-Inf;
    _rep(i,1,n-m){
        int k=read_int();
        while(k--){
            int v=read_int(),w=read_int();
            Insert(i,v,w);
        }
    }
    _rep(i,n-m+1,n)dp[i][1]=read_int(),sz[i]=1;
    dfs(1);
    for(int i=m;i>=0;i--){
        if(dp[1][i]>=0){
            cout<<i;
            break;
        }
    }
}
```

```
    return 0;  
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E5%8A%A8%E6%80%81%E8%A7%84%E5%88%92_3 

Last update: **2021/01/14 20:28**