动态规划选讲

重庆南开中学信息学竞赛教练组

遗留问题

HDU 6981 Rise in Price

在一个$n*n$的网格内，每格有a_{ij}和b_{ij}两个值。从左上角$(1,1)$走到右下角(n,n)，每一步只能向下或者向右走。虽然走法有很多，但都恰好经过了$2n-1$格。设路径上所有的 a_{ij} 之和为 sa，b_{ij} 之和为 sb，求 $sa*sb$ 的最大值。

$1 \le n \le 100$

$1 \le a_{ij}, b_{ij} \le 10^6$

a_{ij}, b_{ij} 均匀**随机生成**

2 3	3 2	1 5	5 1
6 2	3 4	2 3	4 5
3 1	4 2	1 3	4 4
5 4	2 3	4 5	1 3

(2+6+3+2+4+4+1)*

(3+2+4+3+5+4+3)

=22*24

=528

HDU 6981 Rise in Price

- **数据随机**
- 贪心乱搞
  - Wrong Answer

HDU 6981 Rise in Price

- **数据随机**
- 贪心乱搞
  - Wrong Answer
- 暴搜，状态 (sa, sb)
  - 有多少走法?
  - 杨辉三角 C_{2n-2}^{n-1}

1	1	1	1
1	2	3	4
1	3	6	10
1	4	10	20

HDU 6981 Rise in Price

- **数据随机**
- 贪心乱搞
  - Wrong Answer
- 暴搜，状态 (sa, sb)
  - 有多少走法?
  - 杨辉三角 C_{2n-2}^{n-1}

1	1	1	1
1	2	3	4
1	3	6	10
1	4	10	20

- 考虑剪枝
  - 在**同一个位置**，当 sa 和 sb 均不超过另一状态时，不可能发展成最优解
  - dfs 的形式不方便进行这种剪枝
  - 把一个位置的所有方案计算出来再进行剪枝效果更佳

HDU 6981 Rise in Price

- 动态规划
- 设 $dp[i][j][sa]$ 表示从 $(1,1)$ 走到 (i,j) 且 a 的和为 sa 时，sb 的最大值
  - $dp[i][j][sa] = \max(dp[i-1][j][sa - a_{ij}],\ dp[i][j-1][sa - a_{ij}]) + b_{ij}$
  - sa 值域可达 10^8 级别

HDU 6981 Rise in Price

- 动态规划
- 设 $dp[i][j][sa]$ 表示从 $(1,1)$ 走到 (i,j) 且 a 的和为 sa 时，sb 的最大值
  - $dp[i][j][sa] = \max\left(dp[i-1][j]\left[sa - a_{ij}\right],\ dp[i][j-1]\left[sa - a_{ij}\right]\right) + b_{ij}$
  - sa 值域可达 10^8 级别
- 猜测：随机数据下剪枝后有效的 sa 数量较少
  - 用 vector 离散化存储状态 (sa, sb)
- 观察：剪枝后的 (sa, sb) 构成了一个阶梯形状
  - 求 $dp[i][j]$ 相当于合并 $dp[i-1][j]$ 与 $dp[i][j-1]$ 的两个阶梯
  - 两个阶梯的 sa 均有序，可以线性合并（归并）

HDU 6981 Rise in Price

- 数据随机，可本地生成数据观察性质和规律
- 经测试，随机数据下，一个位置的有效状态数量的最大值只能达到2000多一点，平均状态数量只有 300~400，时间复杂度约为 $O(400n^2)$

- 数据随机的标志
  - 题面明示
  - 使用随机数生成器生成数据（线性同余法，$shift - xor$ 法等）

HDU 5838 Mountain [CQOI2012] 局部极小值

在一个 $n*m$ 的网格内填入 $1 \sim n*m$，一个数只能用一次。

定义“山谷”：小于周围 8 连通邻居的值。

求有多少种方案，使得指定的位置‘X’为山谷，其它位置‘.’不为山谷。

$1 \le n,m \le 5$

样例输入

```
2 4
.X..
...X
```

合法方案

```
3 1 4 5
6 7 8 2
```

HDU 5838 Mountain [CQOI2012] 局部极小值

- 数据范围提示我们使用状压DP
  - 2^{25} ?
  - $2^{25} = 33{,}554{,}432$，有亿点大

HDU 5838 Mountain [CQOI2012] 局部极小值

- 数据范围提示我们使用状压DP
  - 2^{25} ?
  - $2^{25} = 33{,}554{,}432$，有亿点大

输入中有8连通相邻的X时，答案为0

- 观察：山谷不能8连通相邻
- 分析：山谷的数量最多只有 9 个
  - 2^{9} 状压山谷的位置是否填写
  - 山谷确定后，周围不能填更小的数
  - 从小到大填数字

```
5 5
X.X.X
.....
X.X.X
.....
X.X.X
```

HDU 5838 Mountain [CQOI2012] 局部极小值

- $dp[i][j]$ 表示填了前 i 个数字，山谷状态为 j 的方案数，接下来填写数字 $i+1$
  - 填入第 k 个 ‘X’ ： $dp[i+1][j|(1 \ll k)] += dp[i][j]$
  - 填入某个 ‘.’ ： $dp[i+1][j] += \text{cnt}[i][j] * dp[i][j]$
  - $\text{cnt}[i][j]$ 表示状态 (i,j) 下剩余可填的 ‘.’ 数量

HDU 5838 Mountain [CQOI2012] 局部极小值

- $dp[i][j]$ 表示填了前 i 个数字，山谷状态为 j 的方案数，填写数字 $i+1$ 转移
  - 填入第 k 个山谷： $dp[i+1][j|(1 \ll k)] \mathrel{+}= dp[i][j]$
  - 填入一个非山谷： $dp[i+1][j] \mathrel{+}= \mathrm{cnt}[i][j] * dp[i][j]$
  - $\mathrm{cnt}[i][j]$ 表示状态 (i,j) 下剩余可填的 ‘.’ 数量
    - $\mathrm{cnt}[i][j] = n*m - S - i + popcount(j)$ ？ （S 为 ‘X’ 总数）
    - 状态 j 下未填的山谷周围此时也不能填
    - $\mathrm{cnt}[i][j]$ 中的 $n*m-S$ 应改为状态 j 下周围没有未填山谷的 ‘.’ 数量
    - $\mathrm{cnt}[i][j]$ 可预处理
  - 时间复杂度 $O(n*m*2^9)$

HDU 5838 Mountain [CQOI2012] 局部极小值

- 这个题做完了吗?
- $5 * 5 * 2^9 = 12800$?

- 刚才的状态设计和转移只保证了 ‘X’ 的位置一定成立
- 但题目要求 ‘.’ 不能为山谷

- 多算了减去就行了
- 容斥!

HDU 5838 Mountain [CQOI2012] 局部极小值

- 通过 dfs 枚举可能成为 ‘X’ 的 ‘.’ 位置，将其修改为 ‘X’ 后再进行一次计算，根据新增 ‘X’ 的数量进行容斥

- 极限情况为5*5的全 ‘.’ 的地图，经测试有6427种合法的 ‘X’ 分布
- 与 2^S 加权后大约为 204,627

- 5*5*204627 = 5,115,675

(换根)树形DP选讲

换根DP的常规流程

① 通常的树形DP（有根树）

第一次dfs，求出每个点子树内的答案

由多个子节点合并出父节点的答案

② 换根DP（无根树）

第二次dfs，求出每个点为根时的答案

计算“父节点新增子树”对子节点答案的贡献

换根DP图解

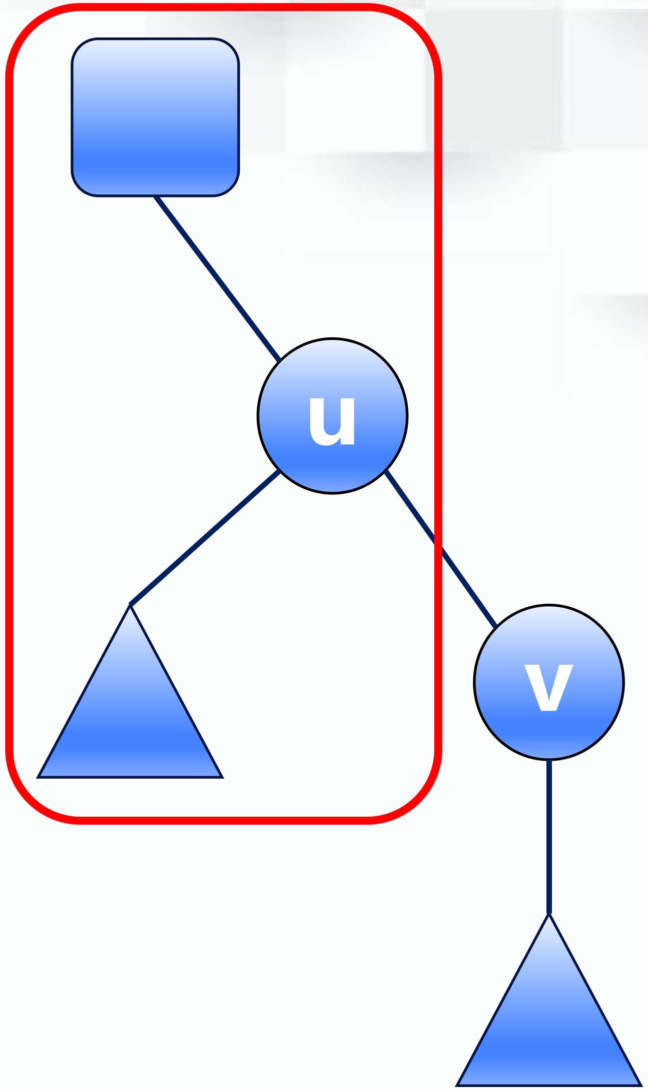
$f[u]$: u子树内的答案

$g[u]$: u为根的答案

第一次dfs算出所有$f[u]$

第二次dfs中，假设已经算出$g[u]$

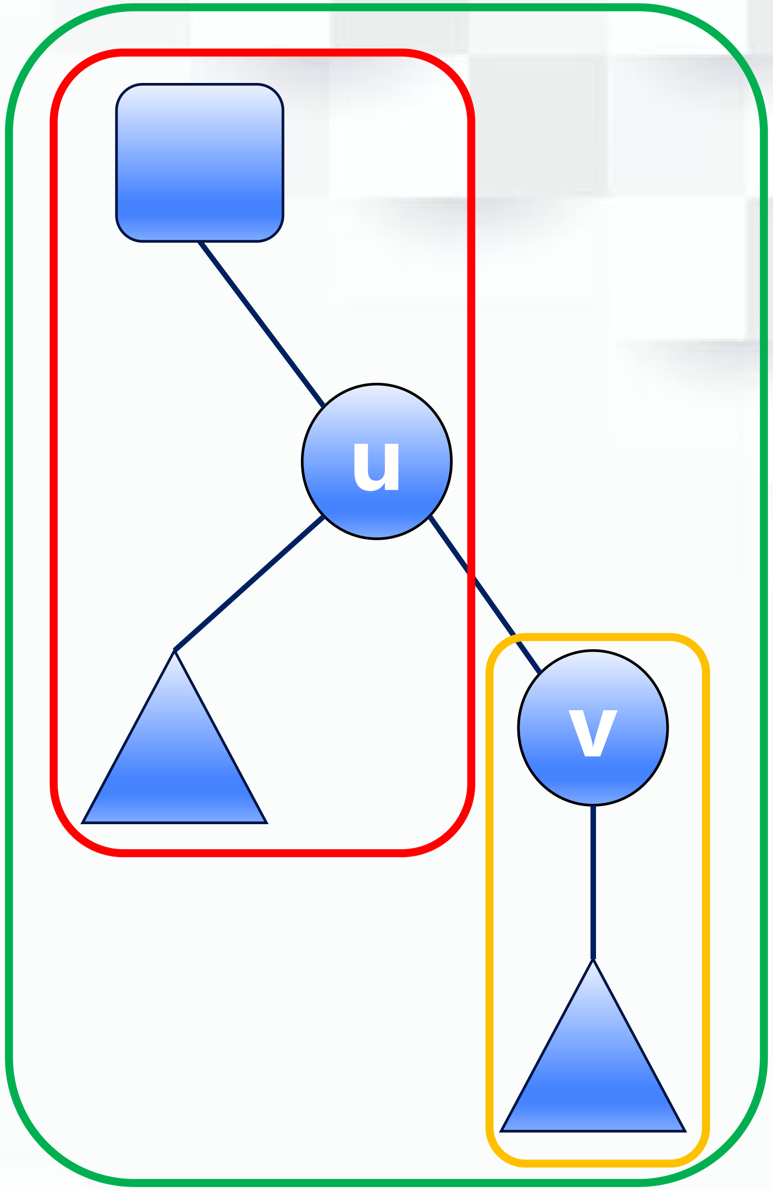
计算$g[v]$相当于在$f[v]$的基础上添加一棵以父节点u为根的子树，如红框所示



换根DP图解

计算红框内子树的答案，关键在于从$g[u]$中去除以v为根的子树 的影响

$$h[v] = g[u] \ominus f[v]$$
$$g[v] = f[v] \oplus h[v]$$



换根DP主要考虑**添加⊕**或**删除⊖**一棵子树对于答案的贡献

换根DP的难点主要在于**删除⊖**操作，即计算$h[v]$

例1 Luogu P3478 [POI2008] STA-Station

给定一个 n 个点的树，请求出一个结点，使得以这个结点为根时，所有结点的深度之和最大。

一个结点的深度之定义为该节点到根的简单路径上边的数量。

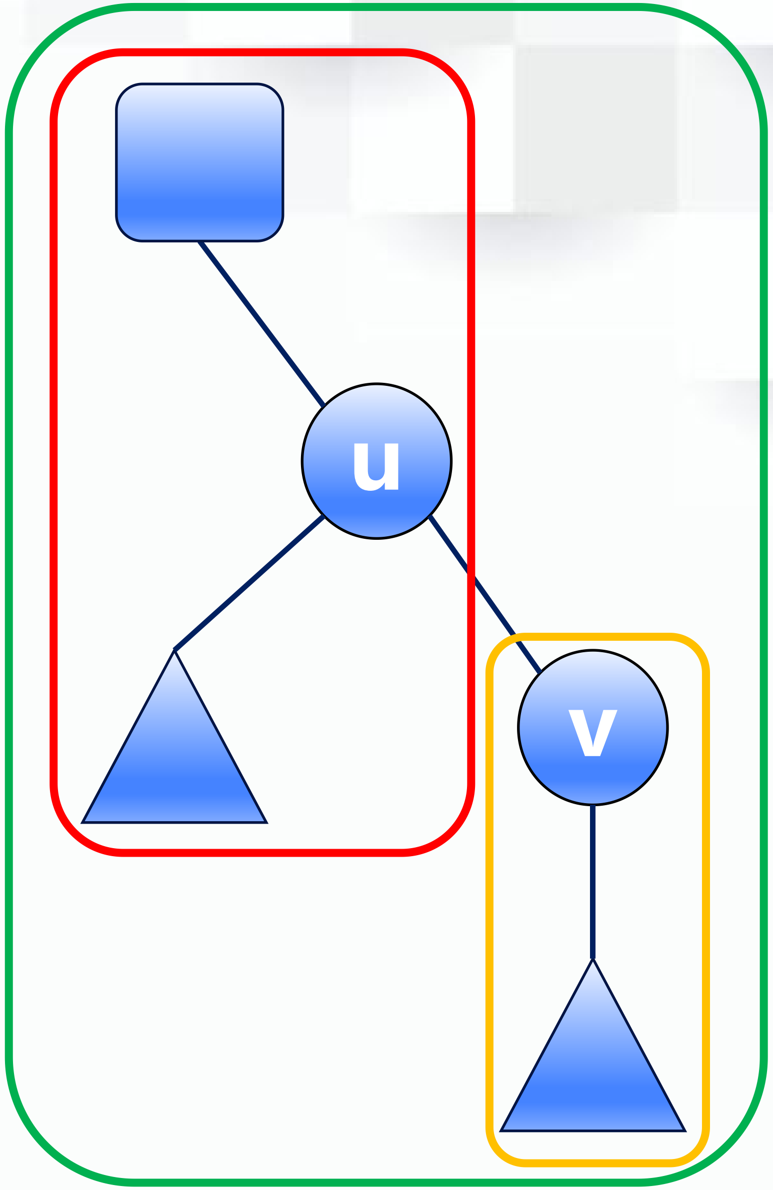
$n \le 10^6$

- 求出每个点为根时的深度之和即可
- 怎么求以1为根，所有子树的答案 $f[u]$?
  - 先dfs1次求出dep，再dfs1次求出子树内dep之和
  - 一次dfs即可，考虑添加一棵子树的贡献，$f[u] = \sum_{v \in Son_u}(f[v] + size[v])$

例1 Luogu P3478 [POI2008] STA-Station

- 以u为根时，子树v 中的节点深度+1
- $g[u] = h[v] + f[v] + size[v]$
- $\Rightarrow h[v] = g[u] - (f[v] + size[v])$

- 以v为根时，子树u 中的节点深度+1
- $g[v] = f[v] + h[v] + (n - size[v])$



例2 POJ 3585 Accumulation Degree

在一棵 n 个点的树中，边有容量。$A(x)$ 定义为：以 x 作为源点，其它度为1的节点作为汇点的最大流。求所有的 $A(x)$ 。

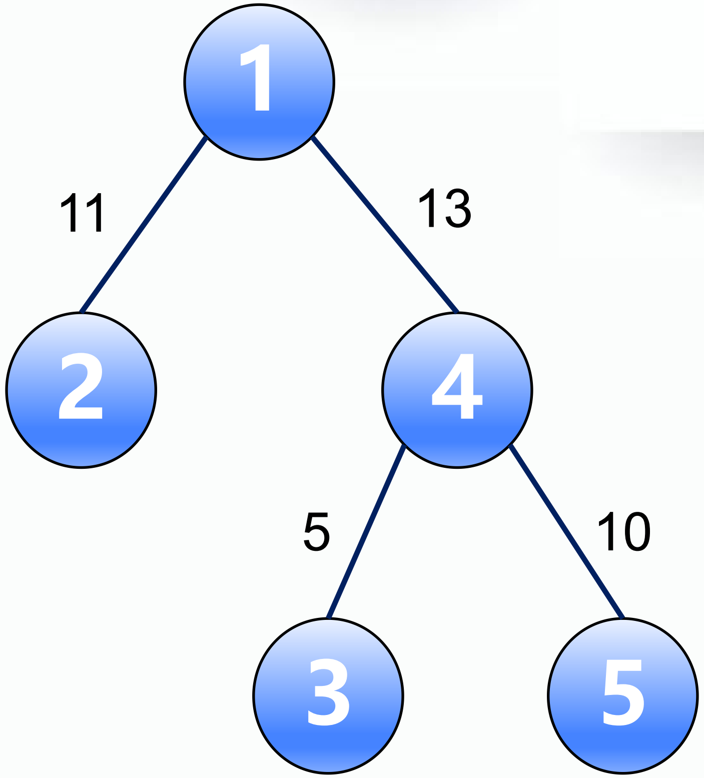
$n \le 2 * 10^5$

- 先考虑固定源点 x ，此时汇点为以 x 为根的有根树中所有的叶子节点
- 直接跑最大流?

- 流量分支不会在中途相交，这看似可以dp
- dp方程怎么列?

例2 POJ 3585 Accumulation Degree

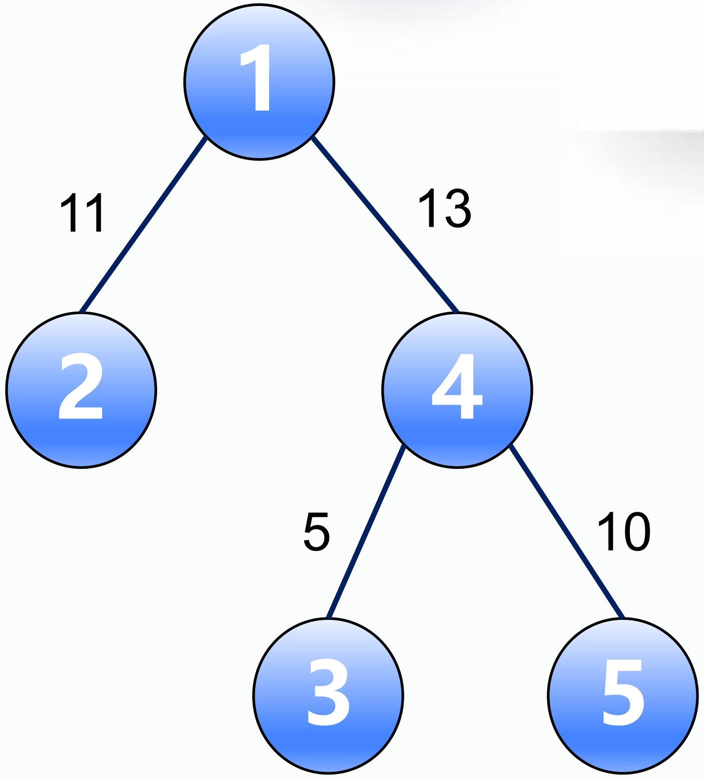
- 在右图的例子中，$f[4]$ 显然为15
- 考虑向 1 中添加 4 这棵子树，能够增加的流量为 $(1,4)$ 这条边的容量 13 和 $f[4] = 15$ 中的较小值
- $f[u] = \sum_{v \in son_u} \min(f[v], c_{uv})$
- 叶子节点直接和汇相连，$f[u]$ 应该初始化为 ∞



例2 POJ 3585 Accumulation Degree

- 换根的转移情况与例1类似，留给大家思考

- 本题还有一个注意点，当换根到原来的叶子节点时，转移与非叶子节点稍有不同



例3 HDU 2196 Computer

在一颗无根树中，边有长度，求每个点为根时的树高。

多组数据，$n \le 10^4$

- 固定根时，你一眼秒了这个水题
- $f[u] = \max\limits_{v \in son_u}(w_{uv} + f[v])$

- 换根时，不好转移
- max 运算不支持“减法”

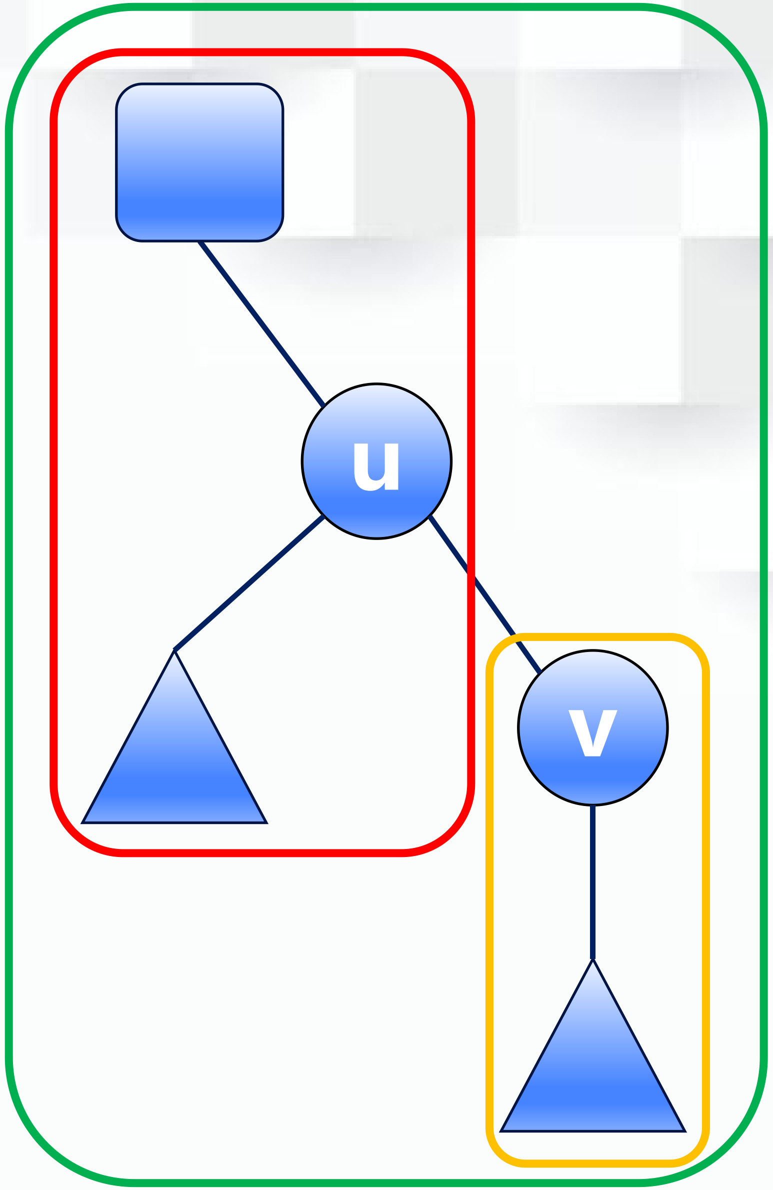
例3 HDU 2196 Computer

- 好在 $f[v] + w_{uv}$ 只是计算 $g[u]$ 的 max 式中的一项

- 若 $f[v] + w_{uv}$ 不是其中最大值，则 $h[v] = g[u]$

- 若 $f[v] + w_{uv}$ 是其中最大值，则去掉它后，$h[v]$ 应该为其中次大值，因此只需要额外维护一下次大值 $f2[u]$ 即可完成转移

- 当转移方程中含有最值函数时，通常需要维护多个最值



例4 HDU 5834 ... with his excited tree

在一棵 n 个点的树中，每个点处有价值为 w_i 的宝物，通过一条边需要支付 c_i 的费用。宝物只能拾取一次，但每次经过一条边都需要支付相应的费用。你想知道以每个点为起点时（终点不固定），能获取的最大收益。

$n \le 10^5$

- 题目描述和数据范围都明示换根DP
- 固定起点时，最优策略一定是：在一些收益为正的子树内往返，最后停留在一棵子树内不返回
- dp 时要计算不返回根的最大收益 $f1[u]$ 和要返回根的最大收益 $f2[u]$

例4 HDU 5834 ... with his excited tree

- 考虑添加子树 v 对于父节点 u 的影响
  - 要返回时，考虑是否有正收益：

$$f2[u] \mathrel{+}= \max(f2[v] - 2c_{uv}, 0)$$

例4 HDU 5834 ... with his excited tree

- 考虑添加子树 v 对于父节点 u 的影响
  - 要返回时，考虑是否有正收益：

  $f2[u] += \max(f2[v] - 2c_{uv}, 0)$

  - 不返回时，计算停留在 v 中相比返回 u 可以获取的收益 p：

    当 $f2[v] - 2c_{uv} \geq 0$ 时，原本会进入 v，则 $p = f1[v] - f2[v] + c_{uv}$

    当 $f2[v] - 2c_{uv} < 0$ 时，原本不进入 v，则 $p = f1[v] - c_{uv}$

  - $f1[u] = f2[u] + \max\{p\}$

- 换根时，$f2[u]$ 是求和，$f1[u]$ 是求 max，都是之前见过的套路了

其它换根相关问题

① 树的重心

- 计算以 u 为根时所有子树size的最大值，使得最大size最小的 u 就是树的重心
- 常规换根dp
- 以父亲为根的子树size为n-size[u]，一次dfs就可以完成

- 性质
  - 重心最多只有2个
  - 重心的子树大小不超过n/2
  - 在树上增加或删除一个叶子节点，重心最多只会移动1条边

其它换根相关问题

② 树的直径

- 即树上最长链

- 两次dfs：先随意选根dfs求出最远点 u，再以u为根dfs找到最远点 v，$u \to v$ 即是一条直径
- 树形dp：求以 u 为根的子树内的最远和次远，相加后全局取max（不用换根）

- 直径合并：两棵树用一条边连接起来，组成一棵新树，则新树某条直径的两个端点一定是在组成原来两树直径的4个点中

练习题

热身训练6

vjudge.net/contest/452738

密码

ilovedp