2020牛客暑期多校训练营（第7场）

陈靖邦、凌浩、陈劲源

牛客竞赛
AC.NOWCODER.COM

惨烈的验题赛

1	300iq# →to practice →devirtualize	9	1409	+ 01:55	+3 01:42	+1 01:21	+ 04:27	+ 00:19	+1 03:27	+1 00:13	+8 01:08		+4 02:57
2	[illegible]	5	602	+ 01:15	+4 04:06	+ 01:08		+ 02:01		+ 00:12			
3	[illegible]	4	600	+2 01:58	+2 04:54	+2 00:43		-10		+ 00:25			
4	[illegible]	3	360	+ 03:06		+1 01:10		-11		+1 01:04			
5	[illegible]	3	446	-2	-6	+ 02:58			+1 03:40	+ 00:28			
6	[illegible]	1	20			+ 00:20				-7			

- D -> Fake News
- I -> Pointer Analysis
- 顺序有一些错位，不要在意

Fake News

- 题意：Σk^2是不是完全平方数。
- 没有人比我更懂数学！！！
- 没有人比我更懂数学！！！
- 没有人比我更懂数学！！！
- 没有人比我更懂数学！！！
- 没有人比我更懂数学！！！
- https://www.zhihu.com/question/363661682

如何证明 $1^2+2^2+\ldots+n^2$ 为平方数的解只有 n=1 或 n=24?

予一人
数学话题下的优秀回答者

HeRaNO 等 112 人赞同了该回答

这不是一个简单的问题，它在数学上小有名气，称为Cannonball problem。1918年被一个叫G.N.Watson 的人用椭圆函数解决，1990年才发现一个初等证明。

https://www.jstor.org/stable/2323911?seq=1
www.jstor.org

发布于 02-01

赞同 112 7 条评论 分享 收藏 喜欢 ···

Fake News

做法0：特判1和24

做法1：打表找规律。

做法1.1：小的范围直接分解，大的范围直接当不行。

做法2：n(n+1)(2n+1)/6，三个乘数分别先两两除下gcd，然后分别判定sqrt是否等于自己就好。

Mask Allocation

- 题意：n*m个口罩，装最少的箱，使得在个数平均的情况下，既能分箱分给n个医院，也能分给m个医院。

- 不妨设（n<m）由于要求字典序最大，所以考虑装口罩最多的盒子，显然不能超过n，不然人数在m的时候这盒子分不出去了。
- 那继续考虑字典序最大，医院数量为n时候需要给每个医院安排m个口罩，我们至多给每个医院安排floor(m/n)个装了n个口罩的盒子，那这里就安排了floor(m/n)*n个盒子，此时每个医院还要额外再拿m%n个口罩才能达到要求。
- 这时候我们考虑医院数量为m的情况，此时需要给每个医院分配n个口罩，那显然已经有floor(m/n)*n个医院满足条件了，还有m%n个医院还啥都没拿到。
- 这样，我们就把问题转化成了n’ = m % n，m’ = n的子问题，不断循环即可。
- 这不就是gcd嘛kora！

Dividing

- 题意: 正整数二元组 Legend Tuple (n, k) 是这样定义的
  - (1, k) 总是 Legend Tuple
  - 若 (n, k) 是 Legend Tuple, 那么 (n + k, k) 也是
  - 若 (n, k) 是 Legend Tuple, 那么 (nk, k) 也是
- 统计有多少个 Legend Tuple (n, k) 满足 $1 \le n \le N$, $1 \le k \le K$, 其中 N 和 K 是不超过 10^{12} 的整数

Dividing

- (n, k) 是 Legend Tuple 当且仅当满足下面三个条件的某一个
  - $n=1$
  - n 是 k 的倍数
  - n-1 是 k 的倍数
- 必要性是显然的
- 我们可以发现若一旦执行过 (n, k) -> (nk, k) 的 Legend Tuple 变动, 那么二元组的第一个整数必然是第二个整数的倍数.
- 若没执行过, 那么 Legend Tuple 必然形如 (xk+1, k)

Dividing

- 充分性简要证明
  - 我们可以认为 (n, k) 是一个 Legned Tuple 当且仅当可以通过将 n 通过减 k 和 除以 k 两种操作变为 1
  - 若 n 是 k 的倍数, 设 n=xk, 我们可以先执行 (x-1) 次 减 k 操作, 使得 n 变成 xk-(x-1)k=k, 最后执行一次除以 k 操作即可
  - 若 n – 1 是 k 的倍数, 设 n=xk+1, 那么执行 x 次除以 k 操作即可
- 做法
  - 先讨论 n 是 k 倍数的情况, 不妨设 n=xk, 那么, x 和 k 中总有一个数字不超过 10^6, 枚举 x 和 k 中较小的一个即可, 另一个的个数可以直接通过计算获取
  - 对于 n=xk+1 的情况也是同样的
  - 对于 n=1 的情况, 直接统计即可

Dividing

- 充分性简要证明
  - 我们可以认为 (n, k) 是一个 Legned Tuple 当且仅当可以通过将 n 通过减 k 和 除以 k 两种操作变为 1
  - 若 n 是 k 的倍数, 设 n=xk, 我们可以先执行 (x-1) 次 减 k 操作, 使得 n 变成 xk-(x-1)k=k, 最后执行一次除以 k 操作即可
  - 若 n – 1 是 k 的倍数, 设 n=xk+1, 那么执行 x 次除以 k 操作即可
- 做法
  - 先讨论 n 是 k 倍数的情况, 不妨设 n=xk, 那么, x 和 k 中总有一个数字不超过 10^6, 枚举 x 和 k 中较小的一个即可, 另一个的个数可以直接通过计算获取
  - 对于 n=xk+1 的情况也是同样的
  - 对于 n=1 的情况, 直接统计即可

牛客竞赛
AC.NOWCODER.COM

Pointer Analysis

- 一个程序中有 26 个对象, 每个对象有 26 个成员指针变量. 同时还有 26 个普通的指针变量. 给定 n 条赋值语句, 询问在以任意顺序执行每条语句无限多次的过程中, 每个指针变量可能指向的对象集合.
- 做法:
- 直接暴力求解即可, 本题难点可能只在于处理读入. 这里给出一个简单的暴力框架, 令 pt(x) 为指针 x 可能指向的对象集合

Pointer Analysis

```
Let worklist be a set
For every allocation statement A = x:
       insert x into pt(A)
       If pt(A) has been changed, add A into worklist
While worklist is not empty:
       While worklist is not empty:
              select one element X from worklist
              delete X from worklist
              For every assignment statement like Y = X:
                     merge pt(X) into pt(Y)
                     If pt(Y) has been changed, add Y into worklist
       For every store statement Y.f = X:
              For every object o in pt(Y):
                     merge pt(X) into pt(o.f)
       For every load statement Y = X.f:
              For every object o in pt(X):
                     merge pt(o.f) into pt(Y)
                      If pt(Y) has been changed, add Y into worklist
```

Pointer Analysis

- Hint by 出题人
- 样例2，就说, B没法指向任何对象, 是空指针.
- 这题的重点是, 要分清对象和指针, 所有的 a, b, c, d, e, f... z 都是对象.
- A, B, C, ..., Z 和 A.a, A.b, ..., A.z 和 o.a, o.b, o.c, ...o.z 这些是指针.

A National Pandemic

题意：一棵树，三种操作：1.一个中心城市x，所有城市y的值+=w-dist(x,y); 2.将城市x的值与0取min 3.询问单点的值。

题解：

2操作是卖萌的，单独记录一个delta就能解决了。

对于1操作，我们考虑一次修改对y来说会增加w-dis(x,y)。

W-dis(x,y)=w-(dep(x)+dep(y)-2*dep(lca))=w-dep(x)-dep(y)+2*dep(lca)

所以，对于每次1操作，我们将其到根上所有点的cnt+=2，询问的时候那部分就是求它到根的权值和。

所以，树上路径加，路径查询，写个qtree改一下输出就好了。

那其实很多人上了点分，复杂度变成一个log，怎么更优就见仁见智了。

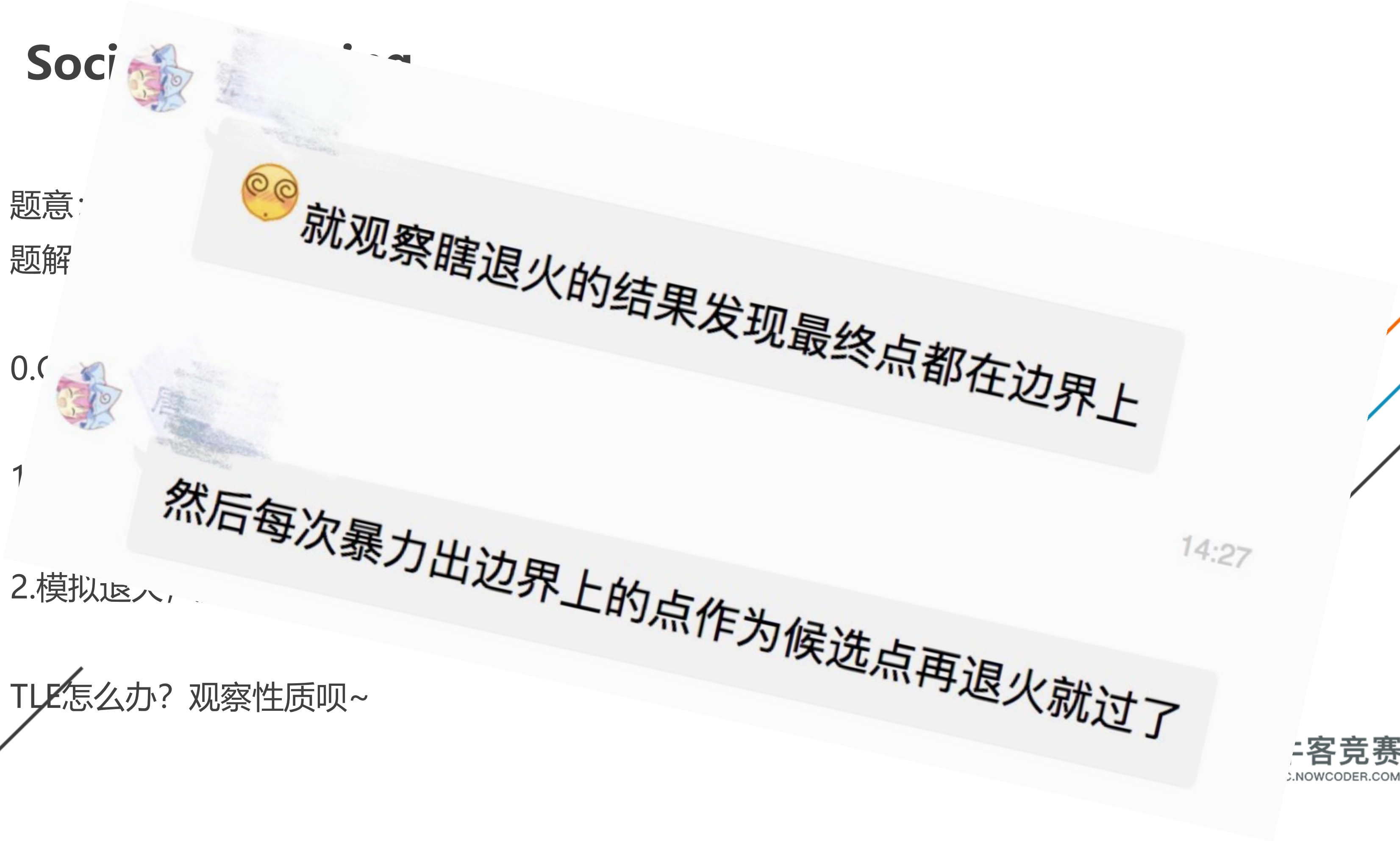
Soci

题意

题解

0.(

1



2.模拟退火

TLE怎么办？观察性质呗~

Social Distancing

Dp也是类似。

整理可得转移方程f[i+1,j+x,k+y]=max[illegible]i,j,k[illegible] i*(x^2+y^2)-[illegible]x-k[illegible]y[illegible]i,j,k]+j^2+k^2) div i[illegible]

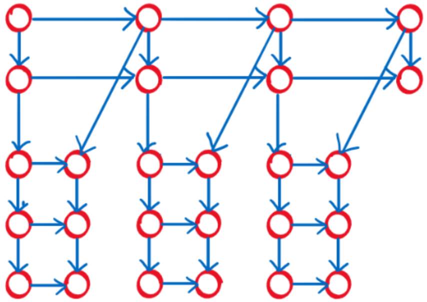
答案为答案为max[illegible]n,j,k[illegible]n[illegible]j[illegible]n[illegible]k[illegible]

由于对于选定的横坐标x选取的点离圆心尽量远更优，可知此时纵坐标y的绝对值为sqrt(r^2-x^2)向下取整，进行DP时只需考虑[illegible]x，y[illegible]abs[illegible]sqrt(r^2-x^2)),-r[illegible]x[illegible]r[illegible]的点即可。

但是还是要打表~

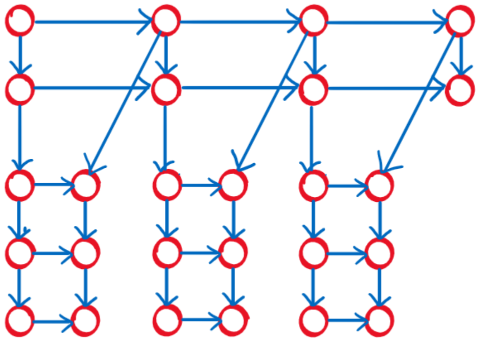
Topo Counting

- 给定一个特殊形态的有向无环图 – 晒肉架图(DRG(n)), 其形态由唯一参数 n 控制, 问 DRG(n) 有多少种拓扑序列.
- 做法:
  - 首先我们需要知道, 如果一个 DAG 是由两个完全不相关的子图组合而成的, 不妨设这两个子图的大小分别是 S_1和 S_2, 对应拓扑序列个数为 P_1和 P_2, 那么这个 DAG 的拓扑序列个数为 $\binom{S_1+S_2}{S_1} P_1 P_2$
  - 以下是 DRG(4) 的形态, 不妨称上面横着的组为架子(V^1), 下面的是肉片($V^2, V^3, \ldots, V^n$).



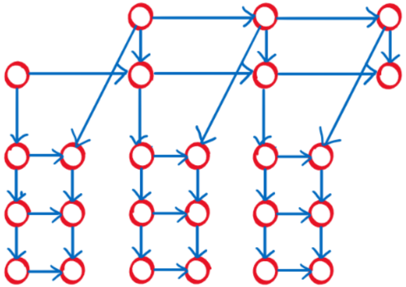
Topo Counting

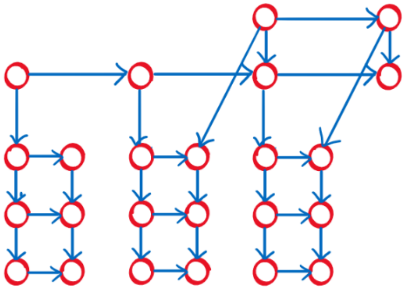
- 拓扑序列实际上就是一个不断删除入度为 0 的节点的过程的删除序列.
- 我们可以发现每个肉片只受制于架子上的两个节点, 当对应的两个节点被删除之后, 肉片就会与整个 DRG 分离. 由于肉片作为一个子图的拓扑序列是非常容易计算的, 所以本题做法的核心就是只维护架子以及连在架子上的肉片的形态.

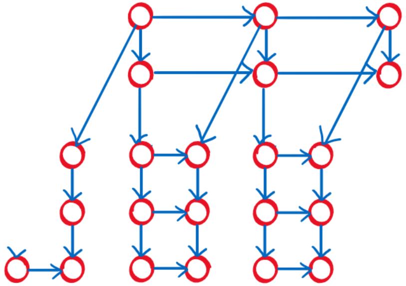


Topo Counting

- 我们可以发现, 在删除节点的过程中, 架子和与架子相连的肉片的整体形态, 只有大致三种情况:
- 第一种是在一个完整的 DRG(n) 中只删除了第一个节点, 不妨设该类形态为 h(n)
- 第二种是在一个完整的 DRG(n) 中删除了第一行中两个以上的连续节点, 这类形态有第一行中剩余节点数 i 和第二行的节点数 j 控制, 不妨设置为 f(i, j)
- 第三种是在一个完整的 DRG(n) 中删除了第一行和第二行的第一个节点, 然后连续删除了第一块肉片的左半边部分, 这类形态由架子上第一行的剩余节点数 i 以及第一块肉片左半边剩余节点数 j 控制, 设置为 g(i, j)

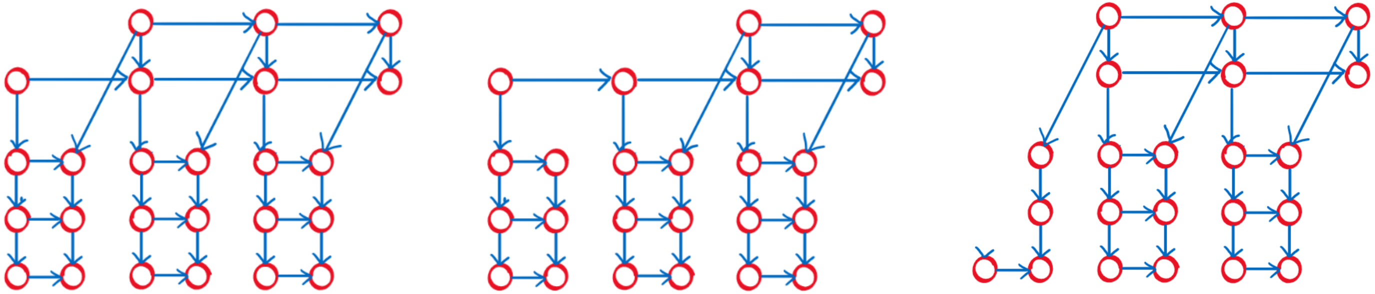






Topo Counting

- 对于 h(n), 接下来只有两种删除方法, 分别会变成 f(n – 2, n) 以及 g(n – 1, n)
- 对于 f(i, j), 接下来也只有两种删除方法, 分别会变成 f(i – 1, j) 以及 (f(i, j – 1) 和 h(j – 1) 中的其中一种+一个分离出去的完整肉片)
- 对于 g(i, j), 同样只有两种情况, 分别变成 g(i, j – 1) 以及 (h(i) + 分离出去的一个肉片)
- 所有这些形态都只有 n^2 种情况, 计算都是常数级, 直接把每个形态的数量都计算出来即可.



Valueable Forests

- 一个森林的代价为内部每个节点度数的平方和.
- 问所有带标号的 n 个点的森林的代价和.
- 做法:
    - 一个森林内部节点的度数平方和等于 2 * (长度为 2 的路径数+长度为 3 的路径数)
    - n 个点的带标号树个数为n^{n-2}
    - 我们可以通过一个 n^2 的做法得到 n 个点的带标号森林个数
    - 我们分别统计长度为 2 的路径的贡献, 相当于从 n 个点里面挑出 2 个点, 设这两个点所在树大小为 j, 那么就需要从剩下 n – 2 个点里面挑出 j – 2 个点, 然后挑出的这 j 个点构成一棵树, 剩下的 n-j 个点构成森林. j 个点构成的树需要以那两个点为根(相当于把这两个点看成一个整体), 用 prufer 序列的处理可以知道方案数是$2j^{j-3}$
    - 长度为 3 的路径同理, j 个点以特定 3 个点为根的方案数是 $3j^{j-4}$

Tokens on the Tree

考虑只有一对(w,b)要怎么处理。不妨假设$w \geq b$，先求出肯定在白色连通块里的点。可以发现，如果存在这样的点集的话，把这些点删掉，剩下连通块大小大于等于b的个数就是答案；否则答案要么是1，要么是2。并且答案为1当且仅当存在一个节点，使得中有两颗子树的$SIZE \geq w$，并存在另一棵子树的$SIZE \geq b$。

根据上面的结论，我们可以很方便的解决这个题。考虑固定一个w，一个节点x一定在白色连通块里，当且仅当节点x的所有子树大小都小于w。当$w = n$的时候，所有点都在白色连通块里。随着w逐渐变小，白色连通块上点个数也逐渐减少，剩下的连通块们会逐渐合并，显然用一个并查集就可以维护这些连通块的大小。

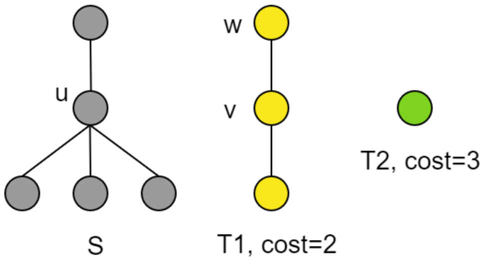
Tokens on the Tree

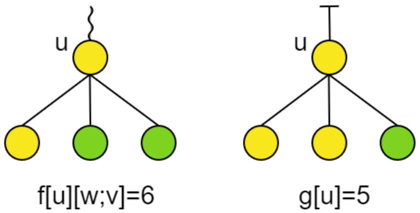
固定一个w，我们需要求的是$w\cdot\sum_{b=1}^{\min(w,n-w)} b\cdot f(w,b)$。在上面用并查集维护连通块大小的同时，我们用线段树或者树状数组维护$s(x)$表示大小超过x的连通块个数，以及$s(x)\cdot x$。那么就能够轻松求出点集存在情况的解。

当白色连通块大小变成0之后，我们需要特判答案是1还是2。注意到可以观察当前有多少节点已经有两个$SIZE\geq w$的子树了，然后从这些节点中找出第三大的子树大小的最大值$bound$，我们根据b和$bound$的大小差来决定哪些b对应的答案是1，哪些b对应的答案是2即可。

NeoMole Synthesis

- 给定一棵树（目标分子）S和另一些树（原料分子）的集合$\{T_i\}$，其中第i个原料分子有一个代价c_i。将S的点集划分为若干连通子块，每个连通子块都与$\{T_i\}$中的一个树同构，要求最小化总的代价。
- 题解：以S中的任意节点为根，将S视为有根树。
  - 对于$st \in E(T_i)$，令记号s;t表示将边st断开后，顶点t所在的子树。
  - 令f[u][s;t]$(u \in S)$表示当u子树中包含子树s;t，u与t对应，且u子树中除与s;t对应的部分外，已经分割成了与$\{T_i\}$中同构的若干连通子图的最小总代价。
  - 令g[u]$(u \in S)$表示将子树u分割成与$\{T_i\}$中同构的若干连通子图的最小总代价。

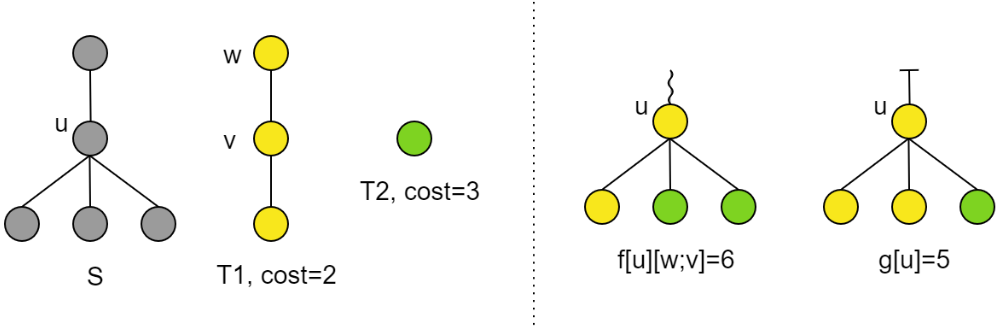




NeoMole Synthesis

- 考虑按照dfs逆序进行树形dp：

```
for u in reversed dfs order of S:
    for every T in {Ti}:
        for every vertex v in T:
            for every vertex w adjacent to v:
                compute f[u][w;v]
            update g[u] with the vertex corresponding to u being v
```



NeoMole Synthesis

- 接下来考虑如何计算f[u][w;v]。对于v的每个相邻节点a（除w外），都要有一个u的子节点b使用状态f[b][v;a]与之匹配；且u的任一子节点最多只能与v的一个相邻节点匹配。
- 更新g[u]的方法类似。只是在考虑a的相邻节点时，无需将w除外。
- 假设v除w外的相邻节点为$a_1, a_2, \cdots, a_n$，u的子节点为$b_1, b_2, \cdots, b_m$，则我们有以下整数规划：

minimize $\sum_{j=1}^{m} c_j$
with respect to $x_{ij} (1 \le i \le n, 1 \le j \le m)$
subject to

1. $\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i$
2. $\sum_{i=1}^{n} x_{ij} \le 1 \quad \forall j$
3. $c_j = \left(1 - \sum_{i=1}^{n} x_{ij}\right) g\left[b_j\right] + \sum_{i=1}^{n} x_{ij} f\left[b_j\right][v; a_i] \quad \forall j$
4. $x_{ij} \in \{0,1\} \quad \forall i, j$

NeoMole Synthesis

minimize $\sum_{j=1}^{m} c_j$
with respect to $x_{ij} (1 \le i \le n, 1 \le j \le m)$
subject to

1. $\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i$
2. $\sum_{i=1}^{n} x_{ij} \le 1 \quad \forall j$
3. $c_j = \left(1 - \sum_{i=1}^{n} x_{ij}\right) g[b_j] + \sum_{i=1}^{n} x_{ij} f[b_j][v; a_i] \quad \forall j$
4. $x_{ij} \in \{0,1\} \quad \forall i,j$

- 将约束3改写为$c_j = g[b_j] + \sum_{i=1}^{n} x_{ij}(f[b_j][v; a_i] - g[b_j])$，则该整数线性规划可转化为带权二部图的最小权完美匹配：二部图的左部有n个点，右部有m个点，第i个左部点与第j个右部点之间的边权为$f[b_j][v; a_i] - g[b_j]$，该二部图的最小权左完美匹配加上$\sum_{j=1}^{m} g[b_j]$即为答案。
- 可使用Kuhn-Munkres算法在$O(n^2 m)$时间内求出答案。
- 总的时间复杂度为$O\left(\sum_{u \in S} \sum_{v \in T_i} \deg(u) \deg(v)^3\right) = O(NM^3)$，最坏情况下仍会超时。

牛客竞赛
AC.NOWCODER.COM

NeoMole Synthesis

- 注意到，对于v的相邻节点为$a_1, a_2, \cdots, a_n$，u的子节点为$b_1, b_2, \cdots, b_m$时的情况，我们实际上要求$\{a_1, a_2, \cdots, a_n\}$与$\{b_1, b_2, \cdots, b_m\}$之间的最小权完美匹配，以及$\{a_1, a_2, \cdots, a_n\}$去掉任一节点后与$\{b_1, b_2, \cdots, b_m\}$之间的最小权完美匹配。
- 我们可以一次性求出全部的最小权完美匹配（而不是调用n+1次Kuhn-Munkres算法）
- 首先，我们调用一次Kuhn-Munkres算法求出$\{a_1, a_2, \cdots, a_n\}$与$\{b_1, b_2, \cdots, b_m\}$之间的最小权完美匹配。
- 要求出去除a_i后的最小权完美匹配，根据网络流理论，只需在上述最小权完美匹配的基础上，找到一条从a_i出发并到达任意右部已匹配节点的权值和最小的交错路*，并沿着交错路退流即可。注意在计算交错路的权值和时，已匹配边的权值需要取负。
- 可使用Floyd-Warshall算法找出所有最短路，并可在总共$O(n^2m)$的时间内求出所有最小权完美匹配。
- 这样，整个算法的复杂度即为$O(NM^2)$。

*: 注意一定存在这样的最短路，因为图中不可能出现负环。若存在负权交错环，将负环上的所有边的匹配/未匹配状态进行切换，则可以得到一个权值和更小的完美匹配，与之前已经求出的最小权完美匹配矛盾。

NeoMole Synthesis

- 解法2：注意到，当T_i中存在大度点时，其所连接的子树大多数是同构的。最坏情况下，也只有$O(\frac{M}{\log M})$个非同构子树。
- 因此我们可以合并所有同构子树，这样最坏情况总复杂度为$O(\frac{NM^3}{\log M})$。
- 由于常数很小，也可以通过本题。

Thanks