

NOIP2018 提高组模拟赛题解

Jingze Xing

2018 年 8 月 1 日

目录

1	Strategy	2
1.1	70 pts	2
1.2	100 pts	2
2	Easy LCA	3
2.1	10 pts	3
2.2	20 pts	3
2.3	30 pts	3
2.4	50 pts	3
2.5	100 pts	3
3	Scarborough Fair	4
3.1	15 pts	4
3.2	50 pts	4
3.3	80 pts	4
3.4	100 pts	4
4	Standard solution: execTime and execMemory	5

1 Strategy

1.1 70 pts

注意到这道题相当于是对 $i = 1 \dots n$ 求不包含物品 i 的背包。所以只要暴力做 n 次背包即可，复杂度 $O(n^3)$ ，这是本场比赛设计的签到分。

1.2 100 pts

由于这种背包是传统的取 \min 的背包，所以只能添加不能删除，CTSC2018Day1T1 的退背包的算法无法奏效。考虑分治法，在处理区间 $[l, r]$ 的答案时，已经将区间 $[1, l-1]$ 和 $[r+1, n]$ 中的物品加入背包。将当前的背包复制一份，然后将 $[mid+1, r]$ 的物品加入背包中，递归处理区间 $[l, mid]$ 的答案。然后将 $[l, mid]$ 的物品加入复制下来的背包中。这样总共 $\log n$ 层，每层中加入背包的物品是 n 个，所以总复杂度是 $O(n^2 \log n)$ 。

这道题的模型来自于 CTSC2018 D1T1 假面，但是通过更改背包的性质，使得分治法成为唯一正确的算法。事实上，该算法在 CTSC2018 赛场上因为常数小也得到了满分。由于此题仅仅是分治法的简单应用，所以适合作为一道不太简单的签到题。

2 Easy LCA

2.1 10 pts

$n \leq 5000$, 此时对于每个左端点暴力往右扫, $lca[l, i] \leftarrow LCA(lca[l, i-1], p_i)$, 使用 $O(1)$ 求 LCA 更新, 可以做到 $O(n^2)$ 。

2.2 20 pts

注意到 LCA 只会向上抬升, 所以每次暴力将 $lca[l, i-1]$ 向上移动直到 p_i 在它的子树中, 这样对于每个左端点复杂度均摊 $O(n)$, 复杂度较小, 常数较小, 经过常数优化可以轻松通过 $n \leq 25000$ 的点。

* 判断一个点是否在另一个点的子树中可以用 DFS 序。

2.3 30 pts

可以发现此时 $lca[l, r] = \min_{l \leq i \leq r} p_i$, 考虑从前向后扫并且维护后缀最小值, 由于 LCA 深度随着后缀长度增长而减小, 所以可以维护一个递增的单调栈, 每次不断把栈顶大于 p_i 的元素拿出来, 最后把这些点压缩成一个带权点加入栈中。均摊复杂度 $O(n)$, 结合上一个算法可以得到 30 分。

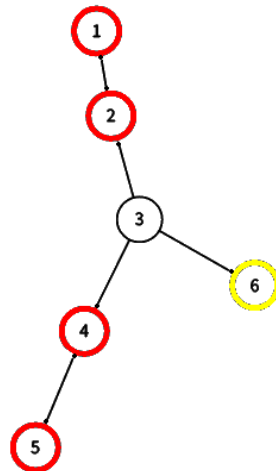
2.4 50 pts

由于随机树深度期望是 $O(\log n)$ 的, 所以对于每一个左端点, 暴力向后二分出 LCA 发生变化的第一个位置, 使用 $O(1)$ 求 LCA 可以做到 $O(n \log^2 n)$ 。

2.5 100 pts

考虑扩展算法 3。如图, 当前后缀的 LCA 一定是在一条链上, 我们在所有后缀后加上一个点时, 相当于把图中 4、5 号点向上抬升到 3 号点。维护一个深度递增的栈, 这里的 3 号点其实就是栈顶的点和新插入点的 LCA, 设该点为 v , 则每次操作不断把栈顶深度超过 $depth[v]$ 的点弹出, 将带权的点 v 和 p_i 重新推入栈中。总复杂度 $O(n \log n)$, 可以通过所有测试点。

此题的模型参考了虚树的建树过程, 是一道思维较灵活, 而算法简单, 代码易实现的 NOIP 中档题, 适合作为 T2。



3 Scarborough Fair

3.1 15 pts

由于 $m \leq 21$ ，所以直接 DFS 枚举每条边的状态，在 DFS 的过程中用并查集合并联通块，可以做到 $O(2^m \cdot \alpha(n))$ 。

3.2 50 pts

这档部分是为状压动规准备的。在 $n = 11$ 时，一个集合（元素不可重）的划分方案数为贝尔数 $Bell(n) = 678570$ 。考虑用最小表示法状压联通状态，则状态数恰好等于 $Bell(n)$ ，依次加入每条边，每条边的转移复杂度在最小表示法上，总复杂度约为 $O(Bell(n) \cdot nm)$ 。由于状态十分不满，所以可以通过该部分测试点。

*** 注意记录状态要使用哈希表，不能用 `std::map`。**

3.3 80 pts

75 分的算法和 100 分算法思路基本上是相同的。考虑分别计算每个极大联通块的贡献，问题转化成计算每个点集联通的概率。按点集大小从小到大依次处理，只需求出这个点集不联通的概率 p ，那么 $1 - p$ 就是点集联通的概率。枚举当前点集 s 中某一个点所在的极大子联通块 t ，那么对 p 产生的贡献即是 $f(t) * P_{t \text{ disconnect } s \setminus t}$ 。枚举 t 中的每个点，计算它和 $s \setminus t$ 不连边的概率。总复杂度 $O(3^n \cdot n)$ 。

3.4 100 pts

上一个算法的瓶颈在于计算两个点集不联通的概率，设 $bet(s)$ 表示整个点集 s 内部所有边都不相连的概率， $inv(s) \equiv bet(s)^{-1} \pmod{998244353}$ ，那么 $P_{t \text{ disconnect } s \setminus t} = bet(s) \cdot inv(t) \cdot inv(s \setminus t)$ 优化为 $O(1)$ 。预处理出 $bet(s)$ 和 $inv(s)$ ，总复杂度 $O(3^n)$ 。

此题考察了状态压缩 DP 的计数技巧，难度较高，部分分较多，区分度高，从多方面考察了选手的计数和 DP 能力，是一道恰到好处的 NOIP T3。

4 Standard solution: execTime and execMemory

```
xing@xing-thinkpad-t440p:~/Documents/A$ g++ strategy.cpp -o strategy -O2
strategy.cpp: In function 'int main()':
strategy.cpp:35:8: warning: ignoring return value of 'int scanf(const char*, ...)', declared here
    scanf("%d%d", &n, &m);
    ~~~~~^~~~~~
strategy.cpp:37:10: warning: ignoring return value of 'int scanf(const char*, ...)', declared here
    scanf("%d%d", &a[i], &b[i]);
    ~~~~~^~~~~~
xing@xing-thinkpad-t440p:~/Documents/A$ time ./strategy < strategy10.in > strategy10.ans

real    0m0.104s
user    0m0.100s
sys     0m0.005s
xing@xing-thinkpad-t440p:~/Documents/A$
```

Status	Success	Time	120ms	Memory	11424KB
--------	---------	------	-------	--------	---------

```
xing@xing-thinkpad-t440p:~/Documents/B$ g++ easy.cpp -o easy -O2
easy.cpp: In function 'int main()':
easy.cpp:44:8: warning: ignoring return value of 'int scanf(const char*, ...)', declared here
    scanf("%d", &n);
    ~~~~~^~~~~~
easy.cpp:46:10: warning: ignoring return value of 'int scanf(const char*, ...)', declared here
    scanf("%d%d", &x, &y);
    ~~~~~^~~~~~
easy.cpp:51:8: warning: ignoring return value of 'int scanf(const char*, ...)', declared here
    scanf("%d", &x);
    ~~~~~^~~~~~
easy.cpp:55:10: warning: ignoring return value of 'int scanf(const char*, ...)', declared here
    scanf("%d", &x);
    ~~~~~^~~~~~
xing@xing-thinkpad-t440p:~/Documents/B$ time ./easy < easy20.in > easy20.out

real    0m0.376s
user    0m0.337s
sys     0m0.036s
xing@xing-thinkpad-t440p:~/Documents/B$
```

```
xing@xing-thinkpad-t440p:~/Documents/C$ g++ fair.cpp -o fair -O2
fair.cpp: In function 'int main()':
fair.cpp:60:8: warning: ignoring return value of 'int scanf(const char*, ...)', declared here
    scanf("%d%d", &n, &m);
    ~~~~~^~~~~~
fair.cpp:62:10: warning: ignoring return value of 'int scanf(const char*, ...)', declared here
    scanf("%d%d%d", &u, &v, &w);
    ~~~~~^~~~~~
xing@xing-thinkpad-t440p:~/Documents/C$ time ./fair < fair20.in > fair20.out

real    0m0.518s
user    0m0.514s
sys     0m0.004s
xing@xing-thinkpad-t440p:~/Documents/C$
```

Status	Success	Time	652ms	Memory	24124KB
--------	---------	------	-------	--------	---------

figure 1: 从上往下依次为 strategy local/xjoi、easy local、fair local/xjoi。