2022.11.18 模拟赛题解

刘胜与

北京大学

2022 年 11 月 18 日

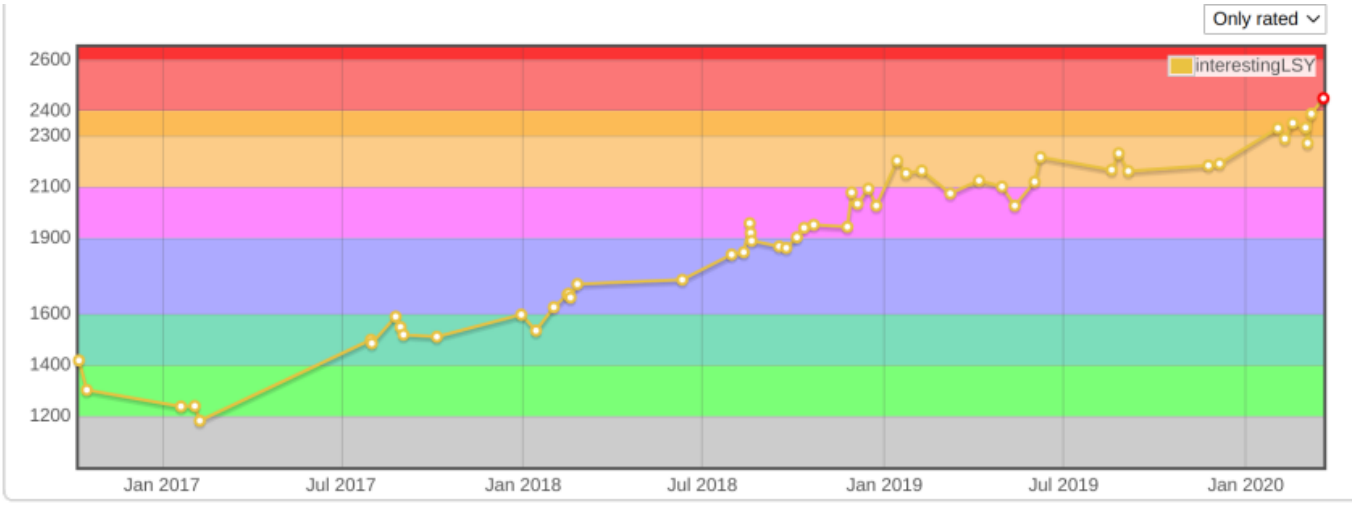
两耳不闻窗外事
一心只读圣贤书

自我介绍

- 刘胜与
- 来自北京大学图灵班
- NOI 2019 银牌，PKUWC2019 一等奖
- Codeforces Grandmaster
- 喜欢开发网站，做过一个 OJ 和一个矿池，也喜欢折腾电脑硬件
- 欢迎大家和我交朋友呀！
- 邮箱地址：interestingLSY@gmail.com

自我介绍

- 刘胜与
- 来自北京大学图灵班
- NOI 2019 银牌，PKUWC2019 一等奖
- Codeforces Grandmaster
- 喜欢开发网站，做过一个 OJ 和一个矿池，也喜欢折腾电脑硬件
- 欢迎大家和我交朋友呀！
- 邮箱地址：interestingLSY@gmail.com



我想和大家说的

- 我出的每道题都不是乱出的。每道题我都会告诉大家，我想向大家传达什么。我认为出题就像写作等其他艺术创作形式，要把自己想传递的东西诉说给选手。每道题我都会标注出，并且在讲题的过程中强调“考点”。

我想和大家说的

- 我出的每道题都不是乱出的。每道题我都会告诉大家，我想向大家传达什么。我认为出题就像写作等其他艺术创作形式，要把自己想传递的东西诉说给选手。每道题我都会标注出，并且在讲题的过程中强调“考点”。
- 我会尽量让题目具有梯度，让每一个层次的选手都能有一个还不错的体验。我尽量保证，只要你没写挂，就能过 1 ～ 2 道题目。

我想和大家说的

- 我出的每道题都不是乱出的。每道题我都会告诉大家，我想向大家传达什么。我认为出题就像写作等其他艺术创作形式，要把自己想传递的东西诉说给选手。每道题我都会标注出，并且在讲题的过程中强调“考点”。
- 我会尽量让题目具有梯度，让每一个层次的选手都能有一个还不错的体验。我尽量保证，只要你没写挂，就能过 1 ～ 2 道题目。
- 我会尽量保证每一道题都让大家有所收获。对于高手，你可以不听前两题的讲解，但我建议你听一下后两道题目的讲解。对于水平偏低的同学，我建议你在理解前两题的基础上，稍微实现以下后两题的部分分。毕竟打暴力也是一种很重要的能力。
  我有个室友就是靠暴力进队，然后又靠打暴力拿了个银牌。

我想和大家说的

- 我出的每道题都不是乱出的。每道题我都会告诉大家，我想向大家传达什么。我认为出题就像写作等其他艺术创作形式，要把自己想传递的东西诉说给选手。每道题我都会标注出，并且在讲题的过程中强调“考点”。
- 我会尽量让题目具有梯度，让每一个层次的选手都能有一个还不错的体验。我尽量保证，只要你没写挂，就能过 1 ～ 2 道题目。
- 我会尽量保证每一道题都让大家有所收获。对于高手，你可以不听前两题的讲解，但我建议你听一下后两道题目的讲解。对于水平偏低的同学，我建议你在理解前两题的基础上，稍微实现以下后两题的部分分。毕竟打暴力也是一种很重要的能力。
  我有个室友就是靠暴力进队，然后又靠打暴力拿了个银牌。
- 我会尽量保证大家学习、复盘、练习的方便性。我不仅会下发标程，也会下发每个部分分对应的程序。

我想和大家说的

- 我出的每道题都不是乱出的。每道题我都会告诉大家，我想向大家传达什么。我认为出题就像写作等其他艺术创作形式，要把自己想传递的东西诉说给选手。每道题我都会标注出，并且在讲题的过程中强调“考点”。
- 我会尽量让题目具有梯度，让每一个层次的选手都能有一个还不错的体验。我尽量保证，只要你没写挂，就能过 1 ～ 2 道题目。
- 我会尽量保证每一道题都让大家有所收获。对于高手，你可以不听前两题的讲解，但我建议你听一下后两道题目的讲解。对于水平偏低的同学，我建议你在理解前两题的基础上，稍微实现以下后两题的部分分。毕竟打暴力也是一种很重要的能力。
  我有个室友就是靠暴力进队，然后又靠打暴力拿了个银牌。
- 我会尽量保证大家学习、复盘、练习的方便性。我不仅会下发标程，也会下发每个部分分对应的程序。
- 如果您比赛体验不好，请尽情在群里吐槽！我希望听到大家的声音！
  当然如果您体验很好，别忘了给个好评！您的肯定对我十分重要。

我想和大家说的

- 我出的每道题都不是乱出的。每道题我都会告诉大家，我想向大家传达什么。我认为出题就像写作等其他艺术创作形式，要把自己想传递的东西诉说给选手。每道题我都会标注出，并且在讲题的过程中强调“考点”。
- 我会尽量让题目具有梯度，让每一个层次的选手都能有一个还不错的体验。我尽量保证，只要你没写挂，就能过 1 ～ 2 道题目。
- 我会尽量保证每一道题都让大家有所收获。对于高手，你可以不听前两题的讲解，但我建议你听一下后两道题目的讲解。对于水平偏低的同学，我建议你在理解前两题的基础上，稍微实现以下后两题的部分分。毕竟打暴力也是一种很重要的能力。
  我有个室友就是靠暴力进队，然后又靠打暴力拿了个银牌。
- 我会尽量保证大家学习、复盘、练习的方便性。我不仅会下发标程，也会下发每个部分分对应的程序。
- 如果您比赛体验不好，请尽情在群里吐槽！我希望听到大家的声音！
  当然如果您体验很好，别忘了给个好评！您的肯定对我十分重要。
- 有问题随时都可以提问！ Questions are welcomed!

题面

intlsy 手上有一个位数为 N 的二进制数字 X，初始情况下 $X = a_0$。
intlsy 会依次进行 M 次操作，第 i 次操作时 intlsy 会令 $X \mathrel{+}= 2^{k_i}$。现在 intlsy 关心两个问题：
1. 每一次操作中，X 中有多少个位 (bit) 发生了变化？
2. 在进行完这 M 次操作后，X 的值为多少？

对于 20% 的数据，有 $N \leq 10, M \leq 100$。
对于 40% 的数据，有 $N \leq 1000, M \leq 2000$。
对于 100% 的数据，$1 \leq N \leq 10^5, 1 \leq M \leq 2 \times 10^5$。

题面

intlsy 手上有一个位数为 N 的二进制数字 X，初始情况下 $X = a_0$。
intlsy 会依次进行 M 次操作，第 i 次操作时 intlsy 会令 $X \mathrel{+}= 2^{k_i}$。现在 intlsy 关心两个问题：
1. 每一次操作中，X 中有多少个位 (bit) 发生了变化？
2. 在进行完这 M 次操作后，X 的值为多少？

对于 20% 的数据，有 $N \le 10, M \le 100$。
对于 40% 的数据，有 $N \le 1000, M \le 2000$。
对于 100% 的数据，$1 \le N \le 10^5, 1 \le M \le 2 \times 10^5$。

考点：模拟，均摊分析。

$N \leq 1000, M \leq 2000$

模拟题目中的操作即可。

$N \leq 1000, M \leq 2000$

模拟题目中的操作即可。
当然，如果你用 python 直接做的话，也是可以的。(虽然 NOIP 上不允许使用 python 但是 OJ 上是可以用的...)

100 分做法

如果你的加法是这么写的：

```
void plus_one(int k) {
    int p = k;
    while (x[p] == 1) {
        x[p] = 0;
        p += 1;
    }
    x[p] = 1;
}
```

那么恭喜你，这个看上去复杂度为 $\Theta(n)$ 的东西，实际执行 m 次以后，复杂度其实是 $\Theta(m)$ 的。

为什么？

均摊分析

的确，每次调用 ‘solve()‘ 的时候，最坏情况下，循环需要循环 n 次。
但是，它不可能每次都循环 n 次啊！
每次调用时，函数会把若干个 1 变成 0，但只会让一个 0 变成 1。
所以，循环的次数的数量级是 $\Theta(m)$ 的！

均摊分析

的确，每次调用 ‘solve()‘ 的时候，最坏情况下，循环需要循环 n 次。
但是，它不可能每次都循环 n 次啊！
每次调用时，函数会把若干个 1 变成 0，但只会让一个 0 变成 1。
所以，循环的次数的数量级是 $\Theta(m)$ 的！

这种“加起来，一起分析，得到一个更紧的上界”的做法就叫“均摊分析”。这就是我想通过这道题传达给大家的。

代码分析

为了照顾低层次的同学，我们简要地分析一下代码。

题意

intlsy 有一个序列 $a_1, a_2, \ldots, a_n$。
对于一个序列 $b_1, b_2, \ldots, b_m$，定义它的权值为 $\sum_{i=1}^{m-1} \max(b_i, b_{i+1})$。
比如，序列 $[1, -2, -3, 4]$ 的权值为
$\max(1, -2) + \max(-2, -3) + \max(-3, 4) = 1 + (-2) + 4 + 3$。
现在 intlsy 想从 $a_1 \ldots a_n$ 的所有子序列中选出权值最大的那个。请你选择一个子序列使其权值最大。请输出对应的权值与对应的子序列。若多个子序列均具有最大的权值，输出任意一个即可。

对于 20% 的数据，有 $n \leq 10, a_i \leq 1000$，
对于 50% 的数据，有 $n \leq 2000, a_i \leq 2000$，
对于另外 10% 的数据，有 $a_i \geq 0$，
对于 100% 的数据，$1 \leq N \leq 2 \times 10^5$，$-2 \times 10^5 \leq a_i \leq 2 \times 10^5$。

题意

intlsy 有一个序列 $a_1, a_2, \ldots, a_n$。
对于一个序列 $b_1, b_2, \ldots, b_m$，定义它的权值为 $\sum_{i=1}^{m-1} \max(b_i, b_{i+1})$。
比如，序列 $[1, -2, -3, 4]$ 的权值为
$\max(1, -2) + \max(-2, -3) + \max(-3, 4) = 1 + (-2) + 4 + 3$。
现在 intlsy 想从 $a_1 \ldots a_n$ 的所有子序列中选出权值最大的那个。请你选择一个子序列使其权值最大。请输出对应的权值与对应的子序列。若多个子序列均具有最大的权值，输出任意一个即可。

对于 20% 的数据，有 $n \leq 10, a_i \leq 1000$，
对于 50% 的数据，有 $n \leq 2000, a_i \leq 2000$，
对于另外 10% 的数据，有 $a_i \geq 0$，
对于 100% 的数据，$1 \leq N \leq 2 \times 10^5$，$-2 \times 10^5 \leq a_i \leq 2 \times 10^5$。

考点：暴力枚举，动态规划（第二个部分分），贪心及相关分析，找规律，开 long long。

先卖个关子

先卖个关子：

你下载大样例了嘛

$n \leq 10, a_i \leq 1000$

暴力枚举每个数字选择与否（共 2^n 种情况），分别计算权值，取最大的那个。

```
vector<int> ansl;
long long ans_weight = 0;
vector<int> selected;
void dfs(int pos) {
    if (pos == N+1) {
        ll sel_weight = calc_weight(selected);
        if (sel_weight > ans_weight) {
            ans_weight = sel_weight;
            ansl = selected;
        }
        return;
    }
    dfs(pos+1);
    selected.push_back(a[pos]);
    dfs(pos+1);
    selected.pop_back();
}
```

复杂度 $O(2^n \cdot n)$。

$n \leq 2000, a_i \leq 2000$

动态规划。设 $dp[i][j]$ 表示在前 i 个数字里选出一个子序列，这个子序列的最后一个数字是 j，此时子序列的最大权值是多少。
想一想转移方程是什么？

$n \leq 2000, a_i \leq 2000$

动态规划。设 $dp[i][j]$ 表示在前 i 个数字里选出一个子序列，这个子序列的最后一个数字是 j，此时子序列的最大权值是多少。
想一想转移方程是什么？

$$dp[i][j] = \begin{cases} dp[i-1][j], & j \neq a[i] \\ max(dp[i-1][j], dp[i-1][k] + max(k, a[i]))\ (\forall k), & j = a[i] \end{cases}$$

复杂度 $O(n \cdot \max\{a_i\})$。

$\forall i, a_i \geq 0$

对于一个序列 $b_1, b_2, \ldots, b_m$，定义它的权值为 $\sum_{i=1}^{m-1} \max(b_i, b_{i+1})$。
所以此时，最优的子序列即为序列本身。直接把序列中所有元素加起来输出，然后输出一遍序列本身就行了。

$\forall i, a_i \geq 0$

对于一个序列 $b_1, b_2, \ldots, b_m$，定义它的权值为 $\sum_{i=1}^{m-1} \max(b_i, b_{i+1})$。
所以此时，最优的子序列即为序列本身。直接把序列中所有元素加起来输出，然后输出一遍序列本身就行了。

$2 \times 10^5 \times 10^5 > 2^{31} - 1$ 是什么？

$\forall i, a_i \geq 0$

对于一个序列 $b_1, b_2, \ldots, b_m$，定义它的权值为 $\sum_{i=1}^{m-1} \max(b_i, b_{i+1})$。
所以此时，最优的子序列即为序列本身。直接把序列中所有元素加起来输出，然后输出一遍序列本身就行了。

$2 \times 10^5 \times 10^5 > 2^{31} - 1$ 是什么？
提醒你会爆 int！这么良心的出题人，上哪里找？(滑稽)

100 分做法

你下载大样例的输出了吗？
如果还没有的话，打开看看，看看你能发现什么？

100 分做法

你下载大样例的输出了吗？
如果还没有的话，打开看看，看看你能发现什么？

最终的答案中，正数和负数好像是交替出现！
为什么？

100 分做法

你下载大样例的输出了吗？
如果还没有的话，打开看看，看看你能发现什么？

最终的答案中，正数和负数好像是交替出现！
为什么？

我们来分析一下：首先，如果我有一个正数没选，那么选上之后子序列权值一定会变大。所以最终的子序列应该包含所有的正数。

100 分做法

你下载大样例的输出了吗？
如果还没有的话，打开看看，看看你能发现什么？

最终的答案中，正数和负数好像是交替出现！
为什么？

我们来分析一下：首先，如果我有一个正数没选，那么选上之后子序列权值一定会变大。所以最终的子序列应该包含所有的正数。
其次，考虑两个正数 A, B：

- 如果 A 和 B 在原序列中就是相邻的，那么没啥好讨论的。
- 如果 A 和 B 在原序列中不是相邻的（他们之间还有其他的负数），那么可以发现，当选出的子序列中，A 和 B 之间恰好有一个负数的时候，答案最优！

这样我们就得到了最终的算法。
当然，别忘了开 long long。

我想传达给大家的

在考试的时候，有时找规律是个不错的选择。
我当年就因为“小凯的疑惑”那题，痛失 40 分。

我想传达给大家的

在考试的时候，有时找规律是个不错的选择。
我当年就因为“小凯的疑惑”那题，痛失 40 分。

常见的找规律方法：

- 观察样例、大样例。

我想传达给大家的

在考试的时候，有时找规律是个不错的选择。
我当年就因为“小凯的疑惑”那题，痛失 40 分。

常见的找规律方法：

- 观察样例、大样例。
- 写一个暴力，给它若干组数据，看看输出是否符合某种规律。

题意

给定一张带权有向图，每个点 i 有三个参数 a_i, b_i, c_i。除了图中原来就有的边之外，如果 $a_i \oplus a_j \geq b_i$ 那么就有一条从 i 到 j 的权值为 $c_i + c_j$ 的有向边。询问点 1 到点 N 的最短路。

对于 10% 的数据，$1 \leq N \leq 10$, $1 \leq M \leq 20$。
对于 30% 的数据，$1 \leq N \leq 1000$, $1 \leq M \leq 2000$。
对于另外 30% 的数据，所有的 b_i 均为 0。
对于 100% 的数据，$1 \leq N \leq 100000, 1 \leq M \leq 200000$。

题意

给定一张带权有向图，每个点 i 有三个参数 a_i, b_i, c_i。除了图中原来就有的边之外，如果 $a_i \oplus a_j \geq b_i$ 那么就有一条从 i 到 j 的权值为 $c_i + c_j$ 的有向边。询问点 1 到点 N 的最短路。

对于 10% 的数据，$1 \leq N \leq 10$, $1 \leq M \leq 20$。
对于 30% 的数据，$1 \leq N \leq 1000$, $1 \leq M \leq 2000$。
对于另外 30% 的数据，所有的 b_i 均为 0。
对于 100% 的数据，$1 \leq N \leq 100000, 1 \leq M \leq 200000$。

算法考点：朴素的最短路，Trie 树，数据结构优化建图。
细节考点：运算符优先级，开 long long，骗分，Spfa vs Dijkstra。
其他：好玩的骗分。

$1 \leq N \leq 1000,\ 1 \leq M \leq 2000$

暴力把图建出来，然后跑最短路（Dijkstra or Spfa）即可。

$1 \leq N \leq 1000,\ 1 \leq M \leq 2000$

暴力把图建出来，然后跑最短路（Dijkstra or Spfa）即可。

注意运算符优先级的问题。在 C++ 里面，等于号（==）的优先级高于异或（^。也就是说，a == b^c 的会被解释为 (a==b)^c。请注意这个问题。

所有的 b_i 均为 0

任意两个 i 和 j，都有一条从 i 指向 j 的权值为 $c_i + c_j$ 的边。
连 N^2 条边是不可行的。
考虑建立一个中间节点 $N+1$，每个节点 i 向 $N+1$ 连一条权值为 c_i 的边，再从 $N+1$ 向 i 连一条权值为 c_i 的边。
这样边数就是 $\Theta(N+M)$ 量级的。

100 分做法

不知道同学们有没有听说过一种技巧叫做“线段树优化建图”？本题我们可以类比它，尝试“Trie 树优化建图”。

具体的，我们开一个 Trie 树，并且把每个点的 a_j 都（按位）插入 Trie 树中。

此时考虑某一个固定的节点 i，其只需要向 Trie 树中的至多 $\log \max\{a_i\}$ 个点连边即可。（有点难描述，看我演示）

100 分做法

不知道同学们有没有听说过一种技巧叫做“线段树优化建图”？本题我们可以类比它，尝试“Trie 树优化建图”。
具体的，我们开一个 Trie 树，并且把每个点的 a_j 都（按位）插入 Trie 树中。此时考虑某一个固定的节点 i，其只需要向 Trie 树中的至多 $\log \max\{a_i\}$ 个点连边即可。（有点难描述，看我演示）

这样的总点数是 $\Theta(N)$ 级别的（想一想为什么不是 $\Theta(N \cdot \log \max\{a_i\})$），总边数是 $\Theta(N \cdot \log \max\{a_i\})$ 的。然后跑 Dijkstra 或 Spfa 即可。
记得开 long long。

骗分！

如果数据是完全随机的，那么边应该会非常密，这样最短路应该只包含几条边。
那我干脆假设最短路只包含两条边（起点 - 中间点 - 终点），并枚举中间点。

骗分！

如果数据是完全随机的，那么边应该会非常密，这样最短路应该只包含几条边。
那我干脆假设最短路只包含两条边（起点 - 中间点 - 终点），并枚举中间点。

在最初版本的数据中，该算法能取得 90 分！
当然，我加强了数据，现在该做法只能取得 25 分。

骗分！

如果数据是完全随机的，那么边应该会非常密，这样最短路应该只包含几条边。
那我干脆假设最短路只包含两条边（起点 - 中间点 - 终点），并枚举中间点。

在最初版本的数据中，该算法能取得 90 分！
当然，我加强了数据，现在该做法只能取得 25 分。

这启发我们，有的时候你可以用出题人意想不到的做法获得大量分数。

拓展：Dijkstra vs Spfa

Dijkstra 的复杂度是正确的。具体复杂度是多少取决于你的数据结构，但一定不会超过 $O((n+m)\log(n+m))$。
Spfa 的复杂度是不一定正确的。在某些情况下（比如网格图），它的复杂度可以变成 $O(nm)$。

拓展：Dijkstra vs Spfa

Dijkstra 的复杂度是正确的。具体复杂度是多少取决于你的数据结构，但一定不会超过 $O((n+m)\log(n+m))$。
Spfa 的复杂度是不一定正确的。在某些情况下（比如网格图），它的复杂度可以变成 $O(nm)$。

所以，Dijkstra 不可能被卡，但 Spfa 有可能。

拓展：Dijkstra vs Spfa

Dijkstra 的复杂度是正确的。具体复杂度是多少取决于你的数据结构，但一定不会超过 $O((n+m)\log(n+m))$。
Spfa 的复杂度是不一定正确的。在某些情况下（比如网格图），它的复杂度可以变成 $O(nm)$。

所以，Dijkstra 不可能被卡，但 Spfa 有可能。

思考题：如何看待这种“卡某个特定算法”的行为？

题意

intlsy 正站在一张网格图上，位于 $(1,1)$，他的目标是到达 (n,m)。intlsy 将从所有从 $(1,1)$ 到 (n,m) 的最短路中等概率随机选择一条路径，并按这条路径从起点走到终点。

intlsy 的口袋中有 a_0 元钱。网格图上有 k 个收费站。每次经过一个收费站时，收费站会从 intlsy 身上收走一块钱，除非他此时已经没有钱了。形式化地，如果在经过收费站前 intlsy 有 X 元钱，那么经过收费站后，intlsy 就会剩下 $\max(X-1,0)$ 元钱。

当 intlsy 到达 (n,m) 后，假设他手上还有 X 元，那么它可以获得 $X^4+3X^3+5^X+6X+233$ 点积分。

请问最后他获得的积分的期望是多少？请输出答案 $\bmod (10^9+7)$ 的值。

对于 10% 的数据，有 $n,m,k\leq 5$。
对于 20% 的数据，有 $n,m,k\leq 15$。
对于 40% 的数据，有 $n,m\leq 250$。
对于另外 20% 的数据，有 $a_0=1$。
对于 100% 的数据，
$1\leq n\leq 100000, 1\leq m\leq 100000, 1\leq k\leq 500, 1\leq a_0\leq 500$。

题意

intlsy 正站在一张网格图上，位于 $(1,1)$，他的目标是到达 (n,m)。intlsy 将从所有从 $(1,1)$ 到 (n,m) 的最短路中等概率随机选择一条路径，并按这条路径从起点走到终点。

intlsy 的口袋中有 a_0 元钱。网格图上有 k 个收费站。每次经过一个收费站时，收费站会从 intlsy 身上收走一块钱，除非他此时已经没有钱了。形式化地，如果在经过收费站前 intlsy 有 X 元钱，那么经过收费站后，intlsy 就会剩下 $\max(X-1,0)$ 元钱。

当 intlsy 到达 (n,m) 后，假设他手上还有 X 元，那么它可以获得 $X^4+3X^3+5^X+6X+233$ 点积分。

请问最后他获得的积分的期望是多少？请输出答案 $\bmod (10^9+7)$ 的值。

对于 10% 的数据，有 $n,m,k\leq 5$。
对于 20% 的数据，有 $n,m,k\leq 15$。
对于 40% 的数据，有 $n,m\leq 250$。
对于另外 20% 的数据，有 $a_0=1$。
对于 100% 的数据，
$1\leq n\leq 100000, 1\leq m\leq 100000, 1\leq k\leq 500, 1\leq a_0\leq 500$。

考点：动态规划，组合计数。

$n, m, k \leq 15$

暴力枚举每一条路径即可。

$n, m \leq 250$

动态规划。
首先我们发现，最终的积分只和“经过了几个收费站”有关。
所以设 $dp[l][i][j]$ 表示从 $(1,1)$ 走到 (i,j)，途径了 l 个收费站的路径总数。
思考一下转移方程？

$n, m \leq 250$

动态规划。
首先我们发现，最终的积分只和“经过了几个收费站”有关。
所以设 $dp[l][i][j]$ 表示从 $(1,1)$ 走到 (i,j)，途径了 l 个收费站的路径总数。
思考一下转移方程？

$$dp[l][i][j] = dp[l'][i-1][j] + dp[l'][i][j-1]$$

其中

$$l' = \begin{cases} l, & (i,j)\text{不是收费站} \\ l-1, & (i,j)\text{是收费站} \end{cases}$$

$n, m \leq 250$

动态规划。
首先我们发现，最终的积分只和“经过了几个收费站”有关。
所以设 $dp[l][i][j]$ 表示从 $(1,1)$ 走到 (i,j)，途径了 l 个收费站的路径总数。
思考一下转移方程？

$$dp[l][i][j] = dp[l'][i-1][j] + dp[l'][i][j-1]$$

其中

$$l' = \begin{cases} l, & (i,j)\text{不是收费站} \\ l-1, & (i,j)\text{是收费站} \end{cases}$$

这样 $dp[l][n][m]$ 即表示从 $(1,1)$ 走到 (n,m) 途径了 l 个收费站的路径数。
复杂度：$\Theta(nmk)$。

引理

首先先来一个比较简单的问题：
从 $(1,1)$ 到 (n,m) 的路径（每次只能向 +X 或向 +Y 方向走）总数是多少？

引理

首先先来一个比较简单的问题：
从 $(1,1)$ 到 (n,m) 的路径（每次只能向 +X 或向 +Y 方向走）总数是多少？

$$\binom{n-1+m-1}{n-1}$$

。以后我们记 $p(x_1,y_1,x_2,y_2)$ 表示从 (x_1,y_1) 走到 (x_2,y_2) 的方案数。

$a_0 = 1$

intlsy 只有一块钱。
我们只要计算出“一个收费站都不途径”的路径条数就行。

$a_0 = 1$

intlsy 只有一块钱。

我们只要计算出“一个收费站都不途径”的路径条数就行。

首先把所有的收费站按照 $x + y$ 排一个序，这样我们最终一定是按排序之后的顺序经过各个收费站的。

$a_0 = 1$

intlsy 只有一块钱。

我们只要计算出“一个收费站都不途径”的路径条数就行。

首先把所有的收费站按照 $x + y$ 排一个序，这样我们最终一定是按排序之后的顺序经过各个收费站的。

设 $dp[i]$ 代表从第 i 个收费站出发，之后不途径任何收费站，到达 (n, m) 的方案数。

思考一下转移方程？

$a_0 = 1$

intlsy 只有一块钱。

我们只要计算出“一个收费站都不途径”的路径条数就行。

首先把所有的收费站按照 $x + y$ 排一个序，这样我们最终一定是按排序之后的顺序经过各个收费站的。

设 $dp[i]$ 代表从第 i 个收费站出发，之后不途径任何收费站，到达 (n, m) 的方案数。

思考一下转移方程？

$$dp[i] = p(x_i, y_i, n, m) - \sum_{j=i+1}^{k} dp[j] \cdot p(x_i, y_i, x_j, y_j)$$

复杂度：$\Theta(k^2)$。

满分做法

让我们来拓展一下上面一个部分分...
设 $f[i][j]$ 代表从收费站 j 出发，经历了（除 j 之外的）i 个收费站，到达 (n, m) 的方案数。
思考一下转移方程？

满分做法

让我们来拓展一下上面一个部分分...
设 $f[i][j]$ 代表从收费站 j 出发，经历了（除 j 之外的）i 个收费站，到达 (n,m) 的方案数。
思考一下转移方程？

当 $i=0$ 时，$f[0][j]=dp[j]$，否则：

$$f[i][j]=p(x_i,x_j,n,m)-\sum_{l=j+1}^{k}p(x_j,y_j,x_l,y_l)\cdot dp[i][l]-\sum_{l=0}^{i-1}dp[l][j]$$

复杂度：$\Theta(k^3)$。

拓展：写组合计数题目的代码的时候的注意事项

- 乘法的时候记得开 long long

拓展：写组合计数题目的代码的时候的注意事项

- 乘法的时候记得开 long long
- 但不要 #define int long long

拓展：写组合计数题目的代码的时候的注意事项

- 乘法的时候记得开 long long
- 但不要 #define int long long
- 及时 mod

拓展：写组合计数题目的代码的时候的注意事项

- 乘法的时候记得开 long long
- 但不要 #define int long long
- 及时 mod
- 但要注意常数

拓展：写组合计数题目的代码的时候的注意事项

- 乘法的时候记得开 long long
- 但不要 #define int long long
- 及时 mod
- 但要注意常数
- 快速幂、求逆等代码的写法不要忘了

祝大家 NOIP RP++!

Thanks!