

A+B Problem 解题报告

湖北省武汉市第二中学 吕凯风

1 试题来源

原创。可以在这里找到：<http://uoj.ac/problem/77>。

2 试题大意

题目名称是吸引你点进来的。

有个 n 个方格排成一行，从左至右依此编号为 $1, 2, \dots, n$ 。要求给这 n 个方格染上黑白两色。

第 i 个方格上有 6 个属性： $a_i, b_i, w_i, l_i, r_i, p_i$ 。

如果方格 i 染成黑色就会获得 b_i 的好看度。如果方格 i 染成白色就会获得 w_i 的好看度。

但是太多了黑色就不好看了。如果方格 i 是黑色，并且存在一个 j 使得 $1 \leq j < i$ 且 $l_i \leq a_j \leq r_i$ 且方格 j 为白色，那么方格 i 就被称为奇怪的方格。

如果方格 i 是奇怪的方格，就会使总好看度减少 p_i 。

现在给你 n, a, b, w, l, r, p ，问所有染色方案中最大的好看度是多少。

设 a_{\max} 为 a, l, r 中的最大值， v_{\max} 为 b, w 中的最大值， p_{\max} 为 p 中的最大值。

编号	n	a_{\max}	v_{\max}	p_{\max}
1	$= 5$	≤ 10	≤ 10	≤ 10
2	$= 20$	≤ 40	≤ 40	≤ 40
3	$= 20$	≤ 40	≤ 40	≤ 40
4	$= 5000$	≤ 10	≤ 200000	≤ 100000
5	$= 5000$	≤ 10	≤ 200000	≤ 300000
6	$= 200$	$\leq 10^9$	≤ 200000	≤ 200000
7	$= 300$	$\leq 10^9$	≤ 200000	≤ 220000
8	$= 500$	$\leq 10^9$	≤ 200000	≤ 400000
9	$= 5000$	≤ 5000	≤ 200000	≤ 150000
10	$= 5000$	$\leq 10^9$	≤ 200000	≤ 300000

时间限制：2s

空间限制：48MB

3 算法介绍

当时我出题的时候觉得这是个绝妙的 idea，应该让更多人知道这道有趣的题目，所以取了这么一个标题党的标题。

3.1 算法一

对于 30% 的数据 $n \leq 20$ 。

于是可以直接 $O(2^n)$ 枚举答案，然后 $O(n^2)$ 求出答案进行更新。

时间复杂度 $O(n^2 \cdot 2^n)$ ，空间复杂度 $O(n)$ 。

期望得分 30 分。

3.2 算法二

对于 30% 的数据 $a_{\max} \leq 10$ 。

于是可以用 DP 来解决。记 $f[i][s]$ 表示考虑前 i 个格子，每个数的状态为 s 的情况下的好看度最大是多少。 s 的第 j 位为 1 表示之前出现了白色的这个数，

否则表示没出现。DP方程很简单就不写了。不过好像内存有点压力，所以用滚动数组搞一下就行了。

时间复杂度 $O(n2^{a_{\max}})$ ，空间复杂度 $O(n + 2^{a_{\max}})$ 。

期望得分 30 分。

3.3 算法三

对于 60% 的数据 $n \leq 500$ 。

对于每个格子有黑白两种状态，可以联想到最小割。一个格子有在 S 割和在 T 割两种状态。

我们来回忆一下最小割与最大流。如果 v 向 u 连一条容量为 w 的有向边，表示 v 如果在 S 割，那么 u 不在 S 割会产生 w 的代价。

一个等价的表述是， u 如果在 T 割，那么 v 不在 T 割会产生 w 的代价。

注意 v 如果在 T 割，那么 u 在 S 割是会产生代价的。

特别的，如果 v 向 u 连一条容量为正无穷大的有向边，表示 v 如果在 S 割，那么 u 一定也要在 S 割。

一个等价的表述是， u 如果在 T 割，那么 v 一定也要在 T 割。

考虑要最大化的式子。发现又有加法又有减法，而我们用的是最小割不是最大割，这样就很坑。

于是可以这样想：对于每个格子，先拿到 b_i 和 w_i 的好看度。然后如果这个格子是黑色，那么付出 w_i 的代价，如果这个格子是白色，付出 b_i 的代价。

这样就全统一成了代价，最小化这个代价即可。

建立结点 $1, 2, \dots, n$ 。结点 i 在 S 割表示格子 i 染黑，在 T 割表示格子 i 染白。

根据前面算代价的方式，那么结点 i 在 S 割需要付出 w_i 的代价，在 T 割需要付出 b_i 的代价。

所以连边：（用 (起始点, 终点, 容量) 表示一条有向边）

- (S, i, b_i)
- (i, T, w_i)

接下来就是奇怪的格子的问题，如果格子 i 是奇怪的格子，就要付出 p_i 的代价。

于是我们希望能有一个结点 i' ，使得如果存在白色的格子 j 满足 $1 \leq j < i$ 且 $l_i \leq a_j \leq r_i$ ，那么 i' 就在 T 割，否则在 S 割。

这样如果构造出来了这样的 i' ，我们希望当且仅当 i 在 S 割， i' 在 T 割时付出 p_i 的代价。于是连边：

- (i, i', p_i)

现在来考虑构造 i' 。

对于每个白色的格子 j 满足 $1 \leq j < i$ 且 $l_i \leq a_j \leq r_i$ ，我们希望 j 如果在 T 割，那么 i' 也一定要在 T 割。于是连边：

- $(i', j, +\infty)$

这样是不是完了呢？其实这样构造出来的 i' ，如果存在白色的格子 j 满足 $1 \leq j < i$ 且 $l_i \leq a_j \leq r_i$ ，那么 i' 一定在 T 割。如果不存在，那么 i' 可能在 S 割也可能在 T 割。

注意到此时 i' 在 T 割肯定比 i' 在 S 割的代价大，由于我们求的是最小割，所以这样构造 i' 是正确的。

如此一来网络流图的点数为 $O(n)$ ，边数为 $O(n^2)$ 。足以通过这 60% 的数据。期望得分 60 分。

3.4 算法四

结合算法二和算法三，可以拿下前 8 个测试点。
期望得分 80 分。

3.5 算法五

为什么算法三不能通过最后两个点呢？原因是边数过多，超出了内存限制。（而且会超出时间限制）

那么有没有办法能减少边的数量呢？答案是有的。

考虑一个这样的问题：现在给你一个染色方案，请在 $O(n \log n)$ 时间内求出这个方案的好看度。

离散化 + 线段树或树状数组的裸题。

既然如此，我们能不能将线段树嵌入到网络流的图里去呢？

首先简化问题，假设没有 $1 \leq j < i$ 这个限制，且所有 a 的值互不相等。

关键是看 i' 的构造。如前所述，我们只要构造出来“如果存在满足条件的 j ，那么 i' 一定在 T 割。如果不存在，那么 i' 可能在 S 割也可能在 T 割”就行了。

考虑一棵线段树，用线段树查询 $[l_i, r_i]$ 这个区间能得到一些结点，这些结点每个代表一个区间，这些不相交的区间的并集就是 $[l_i, r_i]$ 。

那么这些区间里面，只要有一个白色的结点， i' 都必须在 T 割。于是连边：

- $(i', \text{查询到的结点}, +\infty)$

接下来就是保证线段树结点一定满足这样的性质：它所代表的区间中有在 T 割的，那么它也一定在 T 割。于是连边：

- (线段树的非叶结点, 左儿子, $+\infty$)
- (线段树的非叶结点, 右儿子, $+\infty$)
- $(a_i \text{ 对应的线段树叶结点}, i, +\infty)$

这样就成功嵌入进去了。

现在考虑加上 $1 \leq j < i$ 这个限制呢？所有 a 的值仍然互不相等。

只要用函数式线段树就行了，每次在 a_i 这个位置插入，构造出新的线段树。

如果 a 值有相等呢？我们希望满足的是，“它所代表的区间中有在 T 割的，那么它也一定在 T 割。”

于是此时线段树叶结点就要向所有的 k 满足 $k \leq i$ 且 $a_k = a_i$ 的连边了？

显然是不必要的。

只要多加一条这个就行了：

- (当前新建的叶结点, 之前的叶结点, $+\infty$)

如此一来网络流图的点数为 $O(n \log n)$, 边数 $O(n \log n)$ 。

期望得分 100 分。

4 数据生成方式

我想了若干个乱搞的贪心或爬山算法，还有一个写错的网络流算法，然后一个一个地对着卡掉了他们。

5 参考程序

- `color.cpp` 是参考程序。
- `color_orz_orz.cpp` 是一个爆搜。
- `color_orz.cpp` 是一个暴力建图的网络流（用邻接表存图）。
- `color_otl.cpp` 是一个暴力建图的网络流（用邻接矩阵存图）。
- `color_cheater.cpp` 是骗分算法。
- `color_xie_ci_1.cpp` ~ `color_xie_ci_5.cpp` 是 5 个写错了的程序。

6 影响

让我们回到我出这题时间，2013 年。

此前有一场 Codeforces 的 D 题正解是用线段树维护最小费用最大流。去年有一道 NOI 题，前年有一道 POI 题都是在网络流的过程中动态加边。而在刚刚结束的省选中，有一道题是二分图匹配，有一道是最小割。

这些题让我对最小割产生更加深刻的理解。接着某一天，我灵机一动：既然有题目是数据结构能维护最大流，那么可以不可以把数据结构直接放进网络流的建图中呢？于是这道题就产生了。而在出这题的过程中我们仅仅是把最小割当作处理简单逻辑的机器，让我对最小割产生了更加深刻的理解。

恰逢杭州二中的黄嘉泰想在 ContestHunter 上出个比赛，我就把这题放到了那场比赛中。当时这场比赛对外的宣传是：题目很水，还有一道 A + B Problem! 于是吸引了大量不明真相的群众围观，比赛时打开看到题目描述才知道自己被骗了。

当时AC此题的只有柳州高级中学的韦毅龙同学，AC 后惊呼：太丧失了！

最近读到一句话，能产生新的机会的机会才是机会。感觉好题也是如此，能产生新的好题的好题才是好题。我出了这题之后，大家纷纷开始效仿。算法方面，出现了把后缀自动机嵌入网络流、把线段树嵌入图中方便求强连通分量、把树链剖分+线段树嵌入网络流等等。娱乐方面，后来出现了几场比赛所有题目的标题都是什么 A+B Problem, A-B Problem, A*B Problem, A/B Problem 来吸引眼球。

事情已经过了两年，想起来真是一段美好的回忆。如果你问我我到目前为止出过的题中最令我自豪的，我肯定会说：A+B Problem。