

对之前做的codeforces的一些题的总结

题目普遍算法难度不是很高，但是思维难度对我来说就快到头了

希望接下来一年的时间能打上1900分

之前打的比赛也不算少，一次可能做不完，之后会保证每次比赛后都及时总结的

代码就不弄太多了，尽量总结思维方法，如果有重要的实现技巧也会粘上去的

Codeforces round 670

A. Subset Mex

给一堆数，叫你分成两个集合A, B， $Mex(A)$ 表示的是集合A中最小的非负整数，你要最大化 $Mex(A) + Mex(B)$

由于对于一个数来说，如果他是唯一的，那么你丢给A还是丢给B结果是一样的，如果不是唯一的就各分一个，所以把这些数从小到大的丢给A和B即可

B. Maximum product

给一堆数，求出最大的五个数相乘的乘积

先说一下我的做法吧，设 Max_{ij} 表示的是从1到i，最大的j个数的乘积，同理 Min_{ij} 就是最小的j个数的乘积，对于每个数来说，最大的j个数的乘积可能由最大的j-1个数乘上当前的数得来，也可能是由最小的j-1个数乘上当前的数得来，最小的j个数的乘积的转移也同理，这个算法同样适合求最大K个数的乘积

```
1 maxx[j] = max(maxx[j], max(maxx[j - 1] * x, minn[j - 1] * x));
2 minn[j] = min(minn[j], min(minn[j - 1] * x, maxx[j - 1] * x));
```

题解的思路就是先排序，如果最大的5个数的乘积是整数，那就是答案，否则答案就一定是最大的四个数乘剩下n-5个数中其中一个得来的

C. Link Cut Centroids

给一颗树，让你删一条边再加上一条边，使得树的重心变成唯一的

先说重心，如果你把一个节点删去，剩下的所有部分中最大的节点数最小时，那么删去的节点就是重心

重心有个很特殊的性质，就是如果存在两个重心，那么这两个重心一定是相邻的

重心可以用一遍dfs以 $O(n)$ 的时间求下来，如果重心有一个，那么不需要对这个树进行改动

如果重心有两个，删去其中一个重心的一条枝干并把它连接到另一个重心上，即让另一个重心变得重点即可

D. Three Sequences

给一个序列a，叫你分成两个序列b和c，满足 $a_i = b_i + c_i$ 其中b是不降序列，c是不增序列，现在要求最小化序列b,c中的最大值，还附加若干次区间修改，求每次修改后的答案

由于b,c序列的特殊性质，我们所需要最小化的其实是 (c_1, b_n) 有一个比较好的构造方法（虽然我也没想出来）是如果 $a_i < a_{i-1}$ 那么 $c_i = c_{i-1}$ 改变

否则就改变c, 现在令K为 $\sum_{i=2}^n \max(0, a_i - a_{i-1})$ 即所有b的增量, 那么假设 c_1 为x, 那么 b_1 就是 $a_1 - x$, 那么 $b_n = a_n - x + K$ 那么就是最小化 $\max(x, a_1 - x + K)$ 那么x就应该为 $(a_1 + K)/2$

对于q次修改, 使用差分数组维护K的值, 最后O(1)算出答案即可

Educational Codeforces Round 95

A. Buying Torches

一开始有1个木棍, 1个木棍可以换x个木棍, y个木棍换一个煤炭, 1个木棍和1个煤炭可以做一个火把, 问最少换多少次可以做k个火把

正常的算一下就可以得出答案为 $\lceil \frac{ky+k-1}{x-1} \rceil$ 这里有一个实现上的小技巧 $\lceil \frac{a}{b} \rceil = \lfloor \frac{a+b-1}{b} \rfloor$ 注意是整数

B. Negative Prefixes

给一个数组, 一开始有一些数是固定的, 你要重排那些没固定的数, K是使得前缀和小于0的最大的数, 你要最小化K

最直观的想法就是如果总和是负数的话, 那么怎么改变都是没用的, 否则就尽量把正数堆到前面, 这样可以让后面的出现尽可能少的负数

有一个实现细节就是

```
1 | sort(a.rbegin(), a.rend());
```

代表的是把a从大到小排序

C. Mortal Kombat Tower

给一个01序列, A和B分别取数, A先开始, 每次只能取1或者2个, 最小化A取到的1的个数

可以使用 $dp[i][who]$ 作为状态进行转移

另一种思路是如果 $a_1 = 1$ 那么A必须取这个1, 考虑一段连续的1, 我们可以构造一种方案使得第一个1总是可以被B取到, 而对于每一段连续的0, 都可以让A先取到, 对于111, 则B取前两个, A取最后一个, 对于1111, 则B取前两个和最后一个, A取第三个, 对于000, A取第一个和第三个, B取第二个, 那么我们总可以让B先取连续的1, A先取连续的0, 最终计算答案即可。

D. Trash Problem

有n堆垃圾, 每个垃圾有一个坐标, 现在要把这n个垃圾合并成两堆, 问最小的移动步数, 同时有若干个添加垃圾和删除垃圾的操作, 算出每次操作后的最小移动步数

用一个set和一个map, 分别维护所有垃圾的坐标, 以及相邻垃圾的间隔, 那么答案就是最大坐标减去最小坐标再减去最大间隔

对于每次插入和删除操作, 都更新垃圾的坐标以及相邻的间隔, 细节较多