

基础数据结构

单调栈/单调队列

- 很基础，但是思想很有用。
- 单调栈板子:<https://www.luogu.com.cn/problem/P5788>
- 单调队列板子:<https://www.luogu.com.cn/problem/P1886>
- 应用:
 - (1)优化dp，板子:<https://www.luogu.com.cn/problem/P4381>
 - (2)结合线段树(此时单调栈可以认为是低配版笛卡尔树)。
 - (3)减少log。
 - (4)本身结构，比如:<https://www.luogu.com.cn/problem/P8251>。（要扫描线）
 - (5)多重背包(也可以归进(3))。

多重背包

- 这一层转移，如果体积为一，那么就是滑动窗口。
- 然而不是，所以可以划分同余类。

CF372C

- 列出dp。
- 绝对值分两种情况。
- 每层转移使用单调队列。

P4381

- 题意是求基环树直径
- (1)分连通块。
- (2)子树内的直径直接树形dp
- (3)环上考虑破环为链，然后就是标准形式

晾衣服 (shining)

【题目描述】

由于显然今天下大雨，小 pia 的衣服湿了，于是他把衣服都拿去晾。

小 pia 有一根晾衣绳，上面有 n 个用夹子固定好的衣架，第 i 个衣架与第 $i+1$ 个衣架之间间隔了 l_i 厘米 ($1 \leq i < n$)。第 j ($1 \leq j \leq n$) 个衣架下面晾着的衣服的长度为 d_j 厘米。

然而由于众所周知的原因，晾衣绳会不堪重负而下垂，使得衣服掉到地上变得很脏，小 pia 就又要去重新洗一遍，这是小 pia 所不能忍受的。

幸好小 pia 还有一根长为 c 的绳子，可以连接两个衣架来扎紧晾衣绳，小 pia 想知道，加上这条绳子之后是否能让衣服不会掉下来。

具体地，定义晾衣绳的**负担**为任意两件衣服的底部的距离的最大值，距离定义为**两件衣服的长度与两件衣服所在衣架在晾衣绳上的距离之和**，当然，由于新加上的绳子的存在，两个衣架的距离也可能通过这条绳子缩短，你可以看作在一条链上的两点之间连上一条长为 c 的边。（注*：两件衣服不能重合。）

小 pia 想知道将新绳子连接在任意两个衣架之间后晾衣绳的负担的最小值。

【输入格式】

从文件 *shining.in* 中读入数据。

第一行两个整数 n, c ，表示衣架的数量和绳子的长度。

第二行 $n-1$ 个整数 l_1, l_2, \dots, l_{n-1} ，第 i 个整数表示衣架 i 与衣架 $i+1$ 之间的距离。

第三行 n 个整数 d_1, d_2, \dots, d_n ，表示每个衣架上晾着的衣服的长度。

【输出格式】

输出到文件 *shining.out* 中。

一行一个整数，表示答案。

单调栈神题。

- 设连接 $a, b (a < b)$, 二分答案 ans , 考虑 $\{u, v\} (u < v)$ 的相应限制: $X[v] - X[u] + l[u] + l[v] \leq ans$ 或 $|X[u] - X[a]| + |X[v] - X[b]| + l[u] + l[v] \leq ans$
- 后者拆绝对值。令 $F[u] = X[u] - l[u], G[u] = X[u] + l[u]$, 即 $G[v] - F[u] \leq ans$ 或者 $\{G[u] + G[v] \leq ans + X[a] + X[b] \text{ 且 } G[u] - F[v] \leq ans + X[a] - X[b] \text{ 且 } G[v] - F[u] \leq ans + X[b] - X[a] \text{ 且 } -F[u] - F[v] \leq ans - X[a] - X[b]\}$ 让左边最大。
- 即在 $G[v] - F[u] > ans$ 的前提下, 最大化 $\{G[u] + G[v], G[u] - F[v], G[v] - F[u], -F[u] - F[v]\}$. 直接二维数点可以双log, 不过我们希望线性求(毕竟得到不等式之后双指针可以线性, 那么我们希望线性check)
- 枚举 $u, G[v] - F[u]$ 这个限制取 $G[v]$ 越大越好, 此时 $G[u] + G[v]$ 也是 $G[v]$ 越大越好, 所以直接搞一个前缀max。同理, 唯一无法解决的就是 $\max\{G[u] - F[v]\}$ 。

要求: $u < v, G[v] > F[u] + K$; 最大化: $G[u] - F[v]$ 。比较关键的信息是 $G[x] + F[x] = 2X[x]$ 单增。

- 从大到小枚举 u, 维护决策 v 。
- First:: 对于两个点 $v < v'$, 如果 $G[v'] < G[v]$, 那么 $F[v'] > F[v]$, v' 十分劣。(单调栈弹栈)
- Second:: 那么 G 就单调递增了。那么 F 呢? 如果 $v < v', F[v] > F[v']$, 那么我们的 v 确实也不够优秀(直接不加入单调栈)。
- Third:: 实际上, 光上面两条性质还是不够, 因为我们要快速找到 $G[v] > F[u] + k$ 的最小的 v 。不能带 \log , 所以还是有特殊性质。直接分析 u 还是没用的(可以去试试)。

似乎无能为力了。

- 我们有巧妙地观察:考虑 u ,如果它的 F 随着枚举越来越小,那么范围就是逐渐变大的,好办。反之,我也可能在某一个时刻, F 更大了,那么 G 一定更小了,如果不是有限制($u < v$)的话,那么这个 u 就真的没什么用了。这个就对应于一种加入不了栈的情况,即,出现这种情况当且仅当入不了栈。这个时候我只需要讨论和栈顶之间的 v ,而我们知道只有栈顶是有用的。

```
int top=0;mx_b=-INF;
for(int i=1,j=1;i<=n;++i) {
    if(!top||B[i]>=B[st[top]]) {//sth. new
        for(;j>top+1;j--) ;
        for(;j<=top&&A[st[j]]<B[i]-Lim;j++) ;
        if(j-1) chkmin(mi[2],Val-B[st[j-1]]+A[i]);
        for(;top&&A[i]<=A[st[top]];top--) ;
        st[++top]=i;
    }
    else //only think about top
        if(A[st[top]]<B[i]-Lim)
            chkmin(mi[2],Val-B[st[top]]+A[i]);
}
```

P8251(仅限口胡)

- (1)考虑单个贡献，那么左端点没用。
- (2)使用单调栈求计算贡献范围。
- 即:考虑每个点什么时候贡献，则对应于 $l \leq \text{pos}$ 随便， r 应当取一个前缀以确保那个时候仍然在队列，也就是顺着做，弹出时刻-1。
- 相当于矩形加，单点查，也就是单点加，矩形查。就是二维数点。
- (3) 扫描线(当黑盒)

RMQ

- ST表:
 - $O(N\log N)-O(1)$ 。所以完美适配二分，有时候可以用线段树上二分代替。
 - 板子:<https://www.luogu.com.cn/problem/P3865>
- 线段树:
 - $O(N)-O(\log N)$ 。带修。不过它的主要应用并不在RMQ。

P2471 [SCOI2007] 降雨量

- 按题意模拟即可。
- 离散化是难点。

P7596 「EZEC-8」 游戏蛇

RMQ只是工具。

(1)先考虑暴力做法。就是如果向下能赢就向下，否则必须向前继续走。

(2)分开来考虑。找出来左右第一个必胜时刻，谁早谁赢。

(3)讨论必败的情况，取交，两种之中困难的显然是对面进入中间的一个洞。这个时候对方越远（注意，这里不是初始位置）我越容易输。得出来分界点。然后钦定区间里面不合法就是不等式的交，使用rmq。

主要思路:将信息独立。

CF1476F

- 我这个做法有亿点点难写，不过好想。下面就只考虑判断了。如果要写这题还是推荐翻别的题解。
- 考虑前缀， $dp[i]$ 表示前缀 i 里面， i 是最后一个向后cover的，前面没被cover的第一个是什么(以下都需要特别记录满的情况)。这个一定是向后cover的最大的，要不然不优，对吧。
- 然而:转移看起来没法优化，不过可以仔细讨论一下，还是可以做的。
- 提示:和上一题有点像。

一道口胡题。

- A层D1T2读错题版本,以下是原题题面。
- 将题意改为已知最初牌, 求最终字典序最小。

题目描述

题目背景

无论是假插洗还是假切洗似乎都在告诉我们一个事实,“随机”,“shuffle”需要的结构似乎比我们打牌时想象的要复杂。

所以有笨蛋不敢单纯的认为随机 $O(1)$ 次打乱序列产生的就是随机序列,尽管最后证实了他确实是确实个笨蛋。

题目大意

大强面前有 n 叠牌堆, 起初每个牌堆只有一张上面写有数字的牌。

每次大强可以以以下规则**合并相邻两叠牌堆**:

1. $[a_1, a_2, \dots, a_p], [b_1, b_2, \dots, b_q] \rightarrow [a_1, a_2, \dots, a_p, b_1, \dots, b_q]$ 。
2. $[a_1, a_2, \dots, a_p], [b_1, b_2, \dots, b_q] \rightarrow [a_1, b_1, a_2, b_2, \dots, a_p, b_p]$ 这时, 需要保证 $p = q$ 。

经过 $n - 1$ 次合并, 你最终得到了一叠有序牌。

给你**最终**的那一叠牌, 问你**最初**那 n 叠牌, 字典序最小是什么?

输入格式

- 这个拼接没卵用。只考虑交错的操作。
- 一位位填，第一个是固定的，第二个可以取遍一半，并且根据这点我就可以知道很多信息!!(我们不选择点的交错，因为和拼接本质相同)。
- 会把内容划分一下，然后递归下去。
- st表用于寻找决策点。

Huffman树

- k叉板子:<https://www.luogu.com.cn/problem/P2168>

- proof:

补齐之后，结构必然是完全k叉树。考虑交换证明。

有一个gym题，用于解决当叶子节点数量很多，但是大量等价的情况，就是priority_queue的时候有一点不一样，不过我找不到了。

lca

- (1)倍增
- (2)树剖(不会先跳过)
- (3)dfn序/欧拉环游序(前者常数小) :st表求区间深度最小的点。
- 板子:<https://www.luogu.com.cn/problem/P3379>
- 用于提取树链最基本的信息。通常不单独用。
- $\text{lca}(\text{lca}(a,b),c)=\text{lca}(a,\text{lca}(b,c))$ 。

P3398 仓鼠找 sugar

- 求树上两条链是否有交。
- 求出来lca,找到更浅者, 讨论另一条的lca与这一条的两个端点的lca的关系。

树的直径

- 可以DP求,板子:<https://www.luogu.com.cn/problem/P3304>
- 核心性质:树上圆定理->直径的可合并性.(可以得到两边dfs做法),以及中心.

P4408 [NOI2003] 逃学的小孩

- 求 $\max(\min(\text{dis}(c,a), \text{dis}(c,b)) + \text{dis}(a,b))$ 。
- 我们暂且不进行复杂的讨论。
- 考虑三点不共线: 考虑它们的交点, 此时它们处于三个不同子树, dis 取决于到交点的距离, 显然越远越好。显然 a, b, c 中至少一个取到直径的某一个端点。 $\text{ans} = d(a) + d(b) + d(c) + \min(d(a), d(b))$ 显然 c 取端点是不优的, 令 a 取直径端点。
- 分讨, 可以发现 b, c 至少一个在直径另一个端点。显然应该是 b 。
- 至于多条直径: 考虑通过微调单个边的长度来证明任意选一条就行。

CF418D

- 第一步:找到分界点, 然后转化成了连通块内距离某个点最远的点。
- 并且我们这里的连通块的形式是一个子树减去一个子树的形式。
- Sol1:依赖于树上圆定理。把点拍到dfn序上, 线段树维护区间直径。
- Sol2:实际上可以倍增。形式是 $\max::\text{dis}[v]-\text{dis}[\text{lca}]$ 拆成每个边上的问题, 然后查询链上rmq。
- Sol3:一般地, 如果没有想清楚怎么做, 往往考虑树剖, 一些在链上很简单的问题在树剖之后也总能做。

CF1192B

- 题意:单边修改权值, 求动态直径。
- 直径不仅可以往小放缩, 也可往大放缩。
- 需要线段树。
- $\text{dis}(u,v)=d(u)+d(v)-2d(\text{lca})$ 。思路是遍历到答案, 并且如果算错了必须是算的不好, 也就是减多了。
- 欧拉环游序的性质:lca在二者之中, 此外中间的都更深。
- 形如最大三元组ABA, 使用线段树维护之。