line

算法 1

暴力枚举把人分成哪些段，然后计算答案即可。

时间复杂度$O(2^n * poly(n))$，期望得分9分。

算法 2

我们换一种计算答案的方式。设每一段人编号最大的那个是r调程序的时间为t，那么它对答案的贡献是$\sum_{i=r+1}^{n} w_i t$，即不耐烦指数的后缀和。

那么我们可以考虑dp，设f_i表示考虑到第i个人，第i个人在末尾，贡献的最小值。容易写出转移方程。

复杂度$O(n^2)$。

算法 3

一种处理转移方程式中后缀max一项的常见技巧是运用单调栈。它本质上是动态维护了以i结尾的后缀max有哪些段。然后相同的段取f数组维护最小的那个值即可。

而子任务3，4的特征是单调栈中元素个数是期望或严格$O(\log n)/O(1)$的。所以我们可以用RMQ（ST表，线段树等）维护f数组最小的值，并且暴力枚举每个单调栈上的元素，更新答案。

注意到有l_i的限制，所以我们还剩一段不能用单调栈上的元素表示。二分找到那一段，然后特殊处理即可。

当然，对于子任务四，你可以直接使用线段树等数据结构优化dp。

算法 4

注意到我们维护单调栈的入栈，出栈序列可以建成一棵树（类比dfs的过程）。

那么我们每次询问相当于询问树上的一条链组成的所有一次函数在某一点的最小值。

（Q：什么，你说动态凸壳？ A：你来写啊！）

这是一个典型的斜率优化dp的问题。

我们可以先把树建出来，然后做树链剖分。在每条链上维护线段树，线段树上每个节点存储这一段的一次函数组成的凸壳。

注意到斜率的单调性，所以每个线段树上的凸壳可以用单调队列维护。每次对于链上的每个询问，分成的线段树的每个节点得到的值取min即可，这样子避免了凸包合并，动态凸壳等很难实现的操作）。

时间复杂度$O(n \log^2 n)$，期望得分100分。

算法 5

其实我们可以做的更好。我们所需要维护的东西是单调栈的尾端删除，加入，询问一段凸壳上某一点的值。使用二进制分组 + 延迟重构的思想，可以做到$O(n \log n)$

总结

本题作为一道送分题，思维难度不高，考查点是选手熟知的数据结构，动态规划的常见技巧。并且为了增加选手信心，本题也没有刻意地去区分$O(n \log n)$,$O(n \log^2 n)$的算法。这道题目能够给选手更多的时间去攻克后面的难题，给他们一个有力的援助！

time map

简要题意

给定一棵维护按位与的广义线段树，有如下操作：某个节点代表的区间的数与x进行按位与、按位或、按位异或并更新整棵线段树；询问从一个节点开始，按照某种规则往下走并输出经过的节点和。

数据范围：$n \leq 10^6, q \leq 10^5, a_i \in [0, 10^9]$，时间限制为2s，空间限制512MB

子任务 1

该子任务中$n, q \leq 10^3$，暴力维护即可。复杂度为$O(nq)$

子任务 2

该子任务中只有询问操作。可以使用dp得出从每个节点开始往下走的答案。复杂度为$O(n)$

子任务 3

该子任务中只有按位与、按位或修改和询问。

从这里开始，我们需要对题目具体性质进行分析。

考虑如何回答询问。可以观察到，由于该线段树维护的是按位与，那么假设从一个点出发的路径上的数为$a_1, a_2 \dots a_n$，一定有$a_i \& a_{i+1} = a_{i+1} (i < n)$。即这个路径一定能被划分成小于等于$\lfloor \log 10^9 \rfloor$段，每段的数相同。现在问题转化为如何找到这些段之间的分界点。

姑且称题中的广义线段树为J。

我们考虑对J树链剖分。现在可以只考虑如何在一条重链上找这些分界点。因为重链转换最多只会增加$O(\log n)$次切换。我们维护一棵线段树，线段树的每个点存它的**轻儿子**在J上的值，特别的，每条重链的末尾存它本身的值。这样每个点在J的值相当于一条重链上的后缀and和。我们再用线段树二分就可以找到分界点。

然后对于修改，假设现在在修改节点A，那么它影响到的就是A在J上的子树以及祖先。

我们发现，对于子树，可以直接在dfs序上进行区间按位与和区间按位或。这可以通过维护一些标记实现。然后由于我们每个点维护的是**轻儿子**，所以暴力对$O(\log n)$个轻儿子单点修改即可。

时间复杂度$O(n + q_a \log^2 n + q_b \log n (\log 10^9 + \log n))$，其中$q_a$是修改次数，$q_b$是询问次数

子任务 4

该子任务多了异或操作。

这时你会发现，对于一个节点A进行异或操作，并不等同于对于子树的节点进行异或操作。我们必须对该广义线段树进行进一步分析。

考虑每一位。一个J上的节点，把它的区间and和区间or记做一个pair:（and,or），那么一定只会有3种情况：(0,0),(0,1),(1,1)。在线段树上，我们可以考虑维护这些pair的存在性，即对于每一位记3个数表示该区间是否有(0,0),(0,1),(1,1)的点。那么修改操作可以表示为这些数之间的转化。查询也可以用这3种情况的存在性进行查询。

压位实现这些操作即可。

时间复杂度同子任务3。

总结

本题对基本数据结构——线段树进行了拓展，考察了选手对线段树的理解，同时也考察了选手们的代码能力。出题人相信，这道美妙的题目，对你的能力进行了全方面的考察，让你能够认识到自己的不足，一定能给予正在准备北大集训的你，一个有力的援助。

New problem to be configured

题目大意

给一种 HackVM 语言，要求使用该语言实现一些任务。

分点分析

测试点 1

考察选手读题能力。代码为 `rr+p`。

测试点 2

考察选手读题能力。代码为 `rr1^1^/*-p`。

测试点 3

快速幂。为了防止在 HackVM 里写循环，可以使用 C++ 输出 32 次操作。

测试点 4~9

实现对应算法即可。

考虑到对应算法直接在 HackVM 中实现较为困难，且较难调试，可以实现一个转换器，来简化 HackVM 中的操作。

至少需要实现的部分有：`快速 push 大于 9 的数到栈中` 和 `计算 $ 和 g 的位置`。

实现完这两部分之后，应该能较容易的实现测试点 4、5。测试点 6、7 也可能可以实现。

考虑局部变量可以放在操作数栈中，如果进一步实现 `自动管理操作数栈中的变量` 和 `简化的实现条件判断与循环`，可以实现测试点 8、9.

测试点 10

假设现在操作数栈中有一个字符串。如果实现一个算法，能先输出 `在 HackVM 下将该字符串 Push 到操作数栈中的代码`，然后输出该字符串本身。那么把这个算法的字符串放入操作数栈中，然后运行这个算法，得到的就是 Quine 代码。

字符串格式可以是 `前面是字符串每一位的 ASCII 码，最后是这个字符串的长度`。这个算法中不能有 `c` 操作，因为会改变后面的位置。

测试点 11

可以使用 HackVM 自带的操作数栈模拟给定代码的操作数栈。内存和调用栈可以分别指定为某个地址开始的一片内存。

对于每种操作，可以用一个函数执行。如果将内存和调用栈的地址存到内存中一个 ≤ 9 的位置，那么可以在几次操作的时间内实现这些函数。

最后是模拟代码的过程。当前指针可以存到内存中一个 ≤ 9 的位置。代码需要开一片内存存放，但是这片内存的地址也可以存放到内存中一个 ≤ 9 的位置。除此之外，每次判断并调用函数的过程可以不用依次判断，而是直接把对应函数的指针存放到该操作的 ASCII 码指向的内存位置，这样可以大幅减小常数。

总结

本题是一道造计算机的提交答案题，综合考察了选手的算法能力和代码实现能力。相信这道题目能够给选手的 IOI 之路一个有力的援助！

单场总结

本场主要考察了 DP、单调栈、斜率优化、线段树、二进制分组、树链剖分等知识点，三道题的思维难度和代码难度均有一定梯度，部分分设计合理，对集训队水平选手有较好的区分度。出题人相信，这场美妙的练习赛，对你的能力进行了全面的考察，让你能够认识到自己的不足，一定能给予正在准备北大集训的你，一个有力的援助。

line出题人：左骏驰；

time map出题人：袁方舟；

New problem to be configured出题人：王思齐

另：某人的发言