出题人：abruce

T1 世界征服者

设 $f_{i,j}$ 表示前 i 个敌军花了 j 个 a 最少需要多少个 b。朴素 $O(n\,ans^2)$ 转移：
$f_{i,j}=\min\limits_{k=0}^{j} f_{i-1,k}+\lceil\dfrac{h_i-(j-k)a}{b}\rceil$。
考虑如何优化，把这个绝对值式子拆开。我们知道当 $m \bmod k\neq 0\wedge n \bmod k\neq 0$ 的时候 $\lceil\dfrac{m+n}{k}\rceil=\lceil\dfrac{m}{k}\rceil+\lceil\dfrac{n}{k}\rceil-[m \bmod k+n \bmod k\le k]$，于是后面那个绝对值式子就设 $m=h_i-ja,n=ka$。因为 m 可能是负数出现问题，就加一定的 b，后面再减回去。这样就可以用扫描 i 的时候维护以 $n \bmod b$ 为下标的线段树，然后更新 $i+1$ 的 dp 值。需要特殊考虑模 k 为 0 的情况，以及全部用 a 来攻击 h_i 的情况。

T2 悲伤

先考虑 2 操作 $l=r$ 的情况，考虑建一棵线段树。考虑使用标记永久化，这个永久化的标记形成了一个堆。我们再在线段树里维护一个 $maxx$，代表这个区间的最大值。这样，插入和查询操作就很好处理了。

现在我们考虑删除，由于是单点，首先我们找到这个最大值，然后在线段树里找到这个标记。我们需要把这个标记移除，并将其加入这个区间的其它点内，你可以再写一个 pushdown。
时间复杂度 $O(n\log^2 n)$。

接下来考虑优化使其能区间修改。

我们可以想到先找到这一段的最大值是哪一个，然后把标记从这个区间上面移除。这样做看起来时间复杂度是不对的，但我们可以进行剪枝。如果这一个区间的最大值比我们需要删去的小，我们就不去了。乍一看复杂度还是不对，但我们可以分析一下。我们时间复杂度相当于对于每一个最大值的连续区间，把它在线段树上分成不超过 $\log n$ 个区间。由于我们插入的时候是整段一起插入同一个值，所以插入均摊下来会产生 $m\log n$ 个线段树区间。
我们每删除一个产生的线段树区间，最多花 $\log n$ 时间递归下去删除标记。所以删除的时间复杂度为均摊 $O(m\log^2 n)$ （堆的复杂度和向下递归的复杂度是并列的）由于插入和查询的复杂度也是这个量级，所以总复杂度 $O((n+m)\log^2 n)$。

T3 期望

由期望的线性性，我们考虑每一个子集对答案的贡献。它要产生贡献，则它内部联通，且与外部割裂。与外部割裂的概率很好求，我们设 S 内部联通的概率为 f_S 。

对于 f_S，我们考虑容斥。我们将 S 中标号最大的点钦定为新加的点，设 T 为新加点所在连通块。那么 S 内部不连通可转化为 T 内部联通，且 T 与 $C_S T$ 割裂。于是我们可以很方便的 $O(3^n n^2)$ 计算，稍微优化一下可优化到 $O(3^n n)$。

我们还可以进一步优化，考虑 n^2 的部分我们实际上想要得到两个集合之间割裂的概率，于是我们把这两个集合分成前 $\dfrac{n}{2}$ 和后 $\dfrac{n}{2}$ 位，分别计算前 $\dfrac{n}{2}$ 的一个集合 S 和 前 $\dfrac{n}{2}$ 的一个集合 T 之间没有边的概率，前 $\dfrac{n}{2}$ 集合 S 和后 $\dfrac{n}{2}$ 集合 T 之间没有边概率，以此类推，可用 $O(2^n n^2)$ 时间内计算出。于是我们计算 S 与 T 之间的概率就可以把 S 和 T 拆成前 $\dfrac{n}{2}$ 和后 $\dfrac{n}{2}$ 位即可用预处理的东西 $O(1)$ 计算。时

间复杂度优化到 $O(3^n)$ 。

T4 幸终

把那个式子拆开，对于一条祖先-后代链，设 x 为其 d 最小的点，u 为其 d 最大的点，设 $dep_x = mxd - d_x$ 则这条链的代价为 $a_u \times dep_x^2 + b_u \times dep_x + c_u$ ，其中 $a_u = C_u, b_u = 2C_u^2 + C_u, c_u = C_u \times (2C_u + d_u) * (1 - d_u) - 2H_u$ （为了避免出现 0.5，所有东西乘了 2 ）。我们发现这个式子关于 dep_x 单调，即随着 dep_x 的增大，原来不优的决策不会变得更优。（这个可以设一个 Δx 然后拆开式子可证）而我们从底向上更新，我们就只用保存每个子树内关于 dep 最优的决策。为了保存它，我们可以用一个李超线段树进行维护（当然也可以维护二次函数下凸包），插入一个决策的时候类似李超线段树进行分析即可。而合并决策就用李超线段树合并。时间复杂度 $O(n \log n)$ 。