C 搭积木（toybricks）

1. 算法一

我会算 $f(1)$！

求出 a_i 作为最小值的区间 $[L_i, R_i]$，那么高度为 a_i 且包含第 i 列的矩形最大面积为 $a_i \times (R_i - L_i + 1)$，枚举并计算即可。

时间复杂度 $O(n)$，期望得分 1。

2. 算法二

我会算 $f(2)$！

先对 a 建小根笛卡尔树。发现选择的两个矩形要么是两个叠在一起且下面的宽度更大，要么两个在列上无交。

对于第一种情况，等价于在树上选择两个点 u, v 满足 u 是 v 的祖先。设 l_i 为 a_i 子树的区间长度，那么选择 u, v 的答案为 $a_u \times l_u + a_v \times l_v - a_u \times l_v$。李超树合并，在 u 处计入答案即可。

对于第二种情况，可以假定两个矩形不是紧贴着的，否则可以将其归类至第一种情况。这样选择的必定是极长的矩形。找出 $O(n)$ 个极长矩形后即可计算。

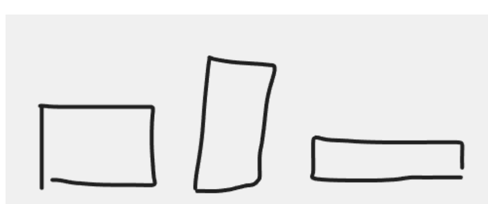
时间复杂度 $O(n \log V)$，结合算法一期望得分 26。

3. 算法三

我会算 $f(3)$！

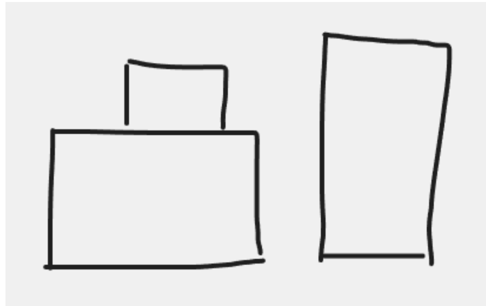
仍然先对 a 建小根笛卡尔树。此时选择的三个矩形的形态有很多种情况，我们来逐个分析。

首先是最简单的情况：三个矩形在列上无交。



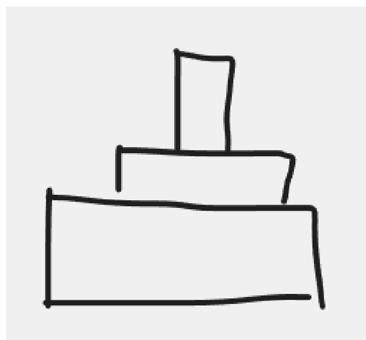
同理找出 $O(n)$ 个极长矩形后 DP 即可。

其次是两个矩形叠在一起，另一个矩形在列上和它们都无交。



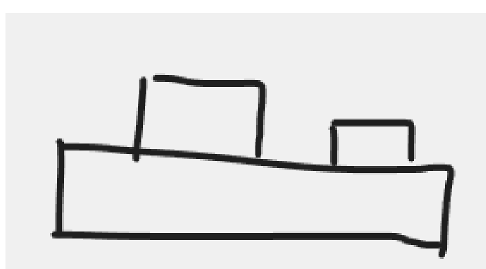
左下的矩形一定是极长的。所以算 $f(2)$ 时可以顺便求出两个量 fl_i 和 fr_i 分别表示以 i 为左端点或右端点的子树 u，在其内部选一个点 v，$a_u \times l_u + a_v \times l_v - a_u \times l_v$ 的最大值。然后即可计算。

然后是三个矩形叠在一起，长度由下往上递减。



类似算 $f(2)$ 一样做就行。就是设 g_u 为在 u 子树内选一个点 v，$a_u \times l_u + a_v \times l_v - a_u \times l_v$ 的最大值。再设 h_u 为在 u 子树内选一个点 v，$a_u \times l_u - a_u \times l_v + g_v$ 的最大值。仍然李超树合并即可。

最后是最难的情况：一个矩形在最下面，另外两个矩形叠在它上面且这两个矩形在列上无交。



这个等价于在树上选择三个点 u, v, w，满足：

- v 和 w 不互为祖孙关系，即 v 和 w 子树区间无交；
- u 为 v 和 w 的共同祖先。

求 $a_u \times l_u + a_v \times l_v + a_w \times l_w - (a_v + a_w) \times l_u$ 的最大值。

发现如果不要求 u 为 v 和 w 的共同祖先，答案不会算得更大。因为若不满足则可以转化为前面讨论过的情况，并且算出来的结果会减多一些项。

所以可以把第二条限制去掉。现在的限制只有 v 和 w 不互为祖孙关系，即 v 和 w 子树区间无交。

若不要求满足这条限制，那么求一个 $(a_i, a_i \times l_i)$ 的凸包，然后和自己做闵可夫斯基和，最后枚举 u 在凸包上二分即可。

这启发我们求出最后的凸包。考虑怎么消除 v 和 w 子树区间无交的限制。

考虑分治。设分治区间为 $[l, r]$，$mid = \left\lfloor \frac{l+r}{2} \right\rfloor$。计算 v 子树区间右端点在 $[l, mid]$ 中，且 w 子树区间左端点在 $[mid+1, r]$ 的答案。

维护两个 $(a_i, a_i \times l_i)$ 的凸包，其中 i 分别满足 i 子树区间左端点或右端点在 $[l, r]$ 内。这样只需要把左右区间得到的凸包做闵可夫斯基和，把新凸包上的所有点扔到最后的点集。

时间复杂度 $O(n(\log n + \log V))$，结合算法一和算法二期望得分 100。