f_i : 前 i 座塔，最少合并几次

p_i : 前 i 座塔，最少合并次数时最后一座合并后的塔高.

for $j = 0 \sim n-1$

for $i = j+1$ to n

if $s_i - s_j \geq p_j$

$f_i \xleftarrow{\min} f_j + (i-j-1)$

$p_i = s_i - s_j$

找到第一个 i，满足 $s_i \geq s_j + p_j$

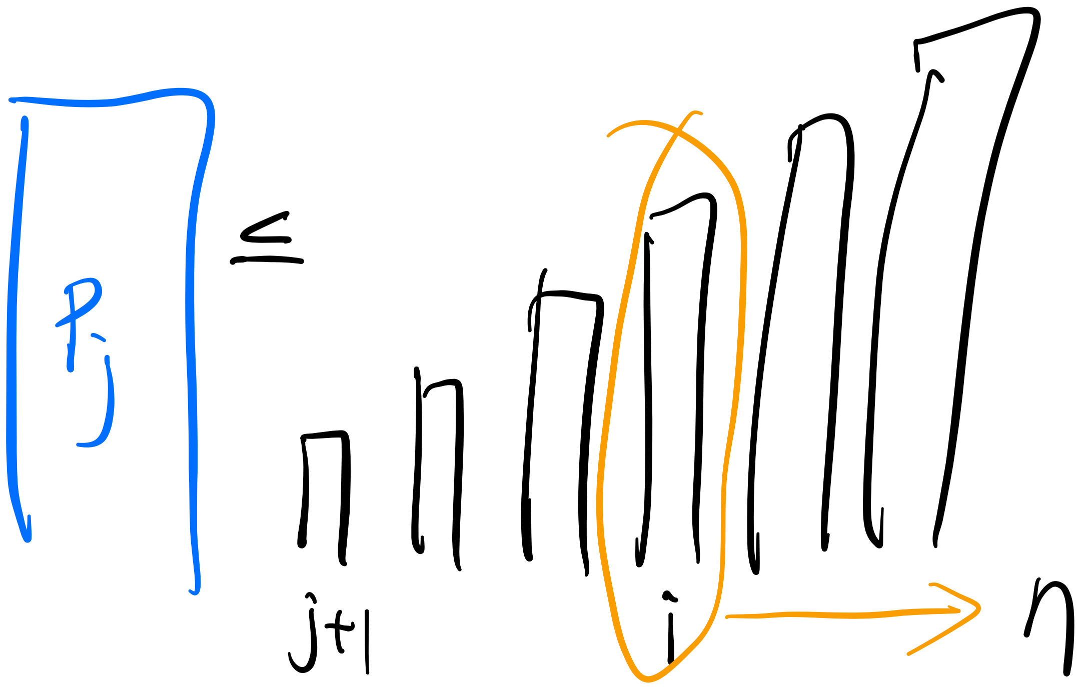
$s_1 \quad s_2 \quad \cdots \quad s_{n-1} \quad s_n$

lower_bound.

$f_i \xleftarrow{\min} f_j + (i-j-1)$ $\quad tag[i] = j$

$f_{i+1} \xleftarrow{\min} f_j + (i+1-j-1)$ $\quad \left. \begin{array}{c} tag[i+1] \\ \vdots \\ tag[n] \end{array} \right\} = j$

还得从第一个 i 循环到 n ???



f_i f_{i+1} ...... f_n

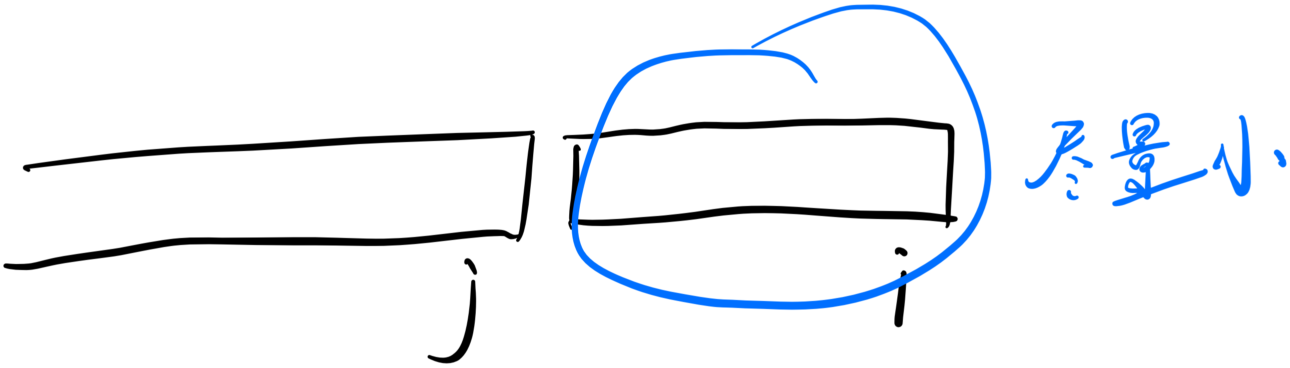
都要由 f_j 更新一次。

除非：有另一个 j'，使得：

用 $f_{j'}$ 更新 f_{i+1} 会让

f_{i+1} 更小。

考虑 f_i 可以由哪些 f_j 更新.



$$s_i - s_j \geq p_j$$

$\Rightarrow$ j 越后越好！

但有可能 j 可以，$j-1$ 又不行.

$\therefore$ 不能二分 j.

$tag[i]$：能用来更新 i 的最后一个 j

若 $tag[i]$ 已知.

$$f_i = f_{tag[i]} + (i - tag[i] - 1).$$

for $j = 0 \sim n$

$tag[j] \xleftarrow{\max} tag[j-1]$

$tag[j]$ 已知.

$f_j = f_{tag[j]} + (j - tag[j] - 1)$

$p_j = s_j - s_{tag[j]}$.

lower_bound 找到第一个可以被 j 更新的 i ($s_i - s_j \geq p_j$

$\Updownarrow$

$s_i \geq p_j + s_j$)

$tag[i] = \cancel{tag[i+1] = \cdots = tag[n]} = j$

$O(n \log n)$.

最小.

长：k.

单调队列：不存无用之物.

有用的东西是单增的.

队头即最小.

窗口向右一位：

① 若队头到窗口外了，弹出队头：O(1)

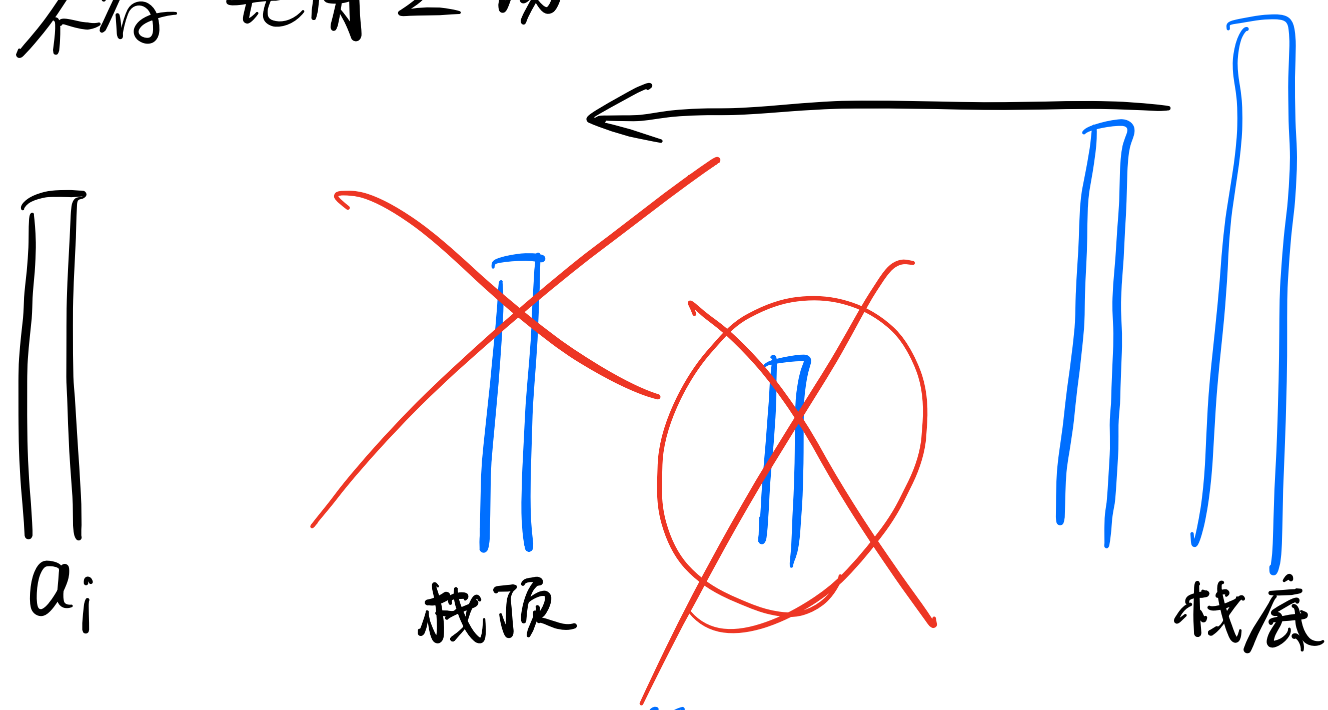
② 不断弹出队尾的"无用之物".

③ 把 a_i 扔进队尾

总共不超过 n 次.

单调栈.

不存无用之物



有用之物单调增.