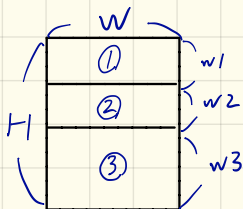


ABC062

C. Chocolate Bar

187-シ 1



で分割

w_1, w_3 をなるべく均等にすれば良い。

H % 3 \Rightarrow 0 差は 0

H % 3 \Rightarrow 1 W_3 だけ大きい

$$w_{\max} - w_{\min} = 1$$

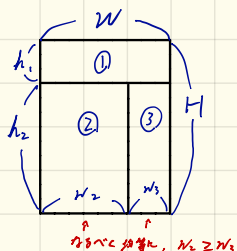
$$H \mid \% 3 \Rightarrow 2 \quad W, \text{ 互換 1 小さい}$$

$$W_{\max} - W_{\min} = 1$$

$\Rightarrow H \% 3 == 0$ と else を考えれば充分

$$h_1$$
 を $\text{range}(1, H//2+1)$ まで動かしたときに

$\max(s_1, s_2, s_3) - \min(s_1, s_2, s_3)$ の
最小値を求む。



のようにT字で
分割

パターン3, 4 ... パターン 1, 2 を90度回転させたもの

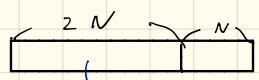
→ try - divi(w, h)

↑↑ 入れかえで 入力すれば良い。

⇒ パターン1~4で1番小さい22が答え。

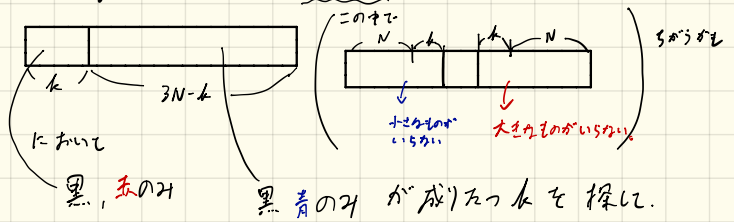
D: 3N Number

(8, 2, 2, 7, 4, 6, 5, 3, 8) が最適



つまり 2N 個入るのが最適な場合もある。

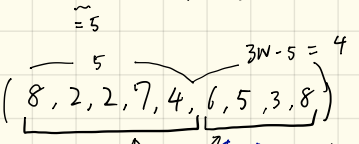
答えを出した時 k in range($N, 2N+1$) で全探索



黒、赤の2
において

黒、青の2 が成った k を探して。

ある k について考えてみる その中で (前 - 後) が最大のものが答え



赤が $N=5$ のとき 青が $N=4$ のとき

赤の合計が大きい方が N とこれより良い but k in range($N, 2N+1$) でテスト

$O(N \times N \log N)$ で効率がある。

→ 優先度付きキューを用いる。

- ・ 要素の push
- ・ 最小要素の pop
- ・ Python では heapq という実装

具体的に、

初期: $k=N$
(8, 2, 2, 7, 4, 6, 5, 3, 8)

$q = \text{queue}$ に push しておく。 $\text{sum_red} = []$
 $\text{sum_red.append}(\text{sum}(q))$

$k+1$: $k=4$
(8, 2, 2, 7, 4, 6, 5, 3, 8)

$q.push(7)$

$q.pop()$

中から前の k 個の中で常に大きい N 個が入る。

$k \leq N$ まで繰り返す。

青の数も同様にして sum_blue を作る

$\max(\text{sum_red} - \text{sum_blue})$
 $\text{sum_red.append}(\text{sum_red}[-1] - \text{pop}(\text{red}) + A[k])$ 要素差
が答え。