S261 - Tribal Occupation

Daniel Hsieh {QwertyPi}

2026-02-14

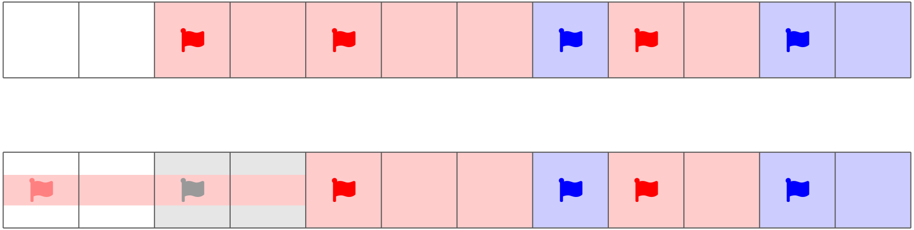
Table of Contents

1. The Problem
2. Subtask 1-3: Building Observations
3. Subtask 4-5: Efficient Exhaustions
4. Full Solution

Background

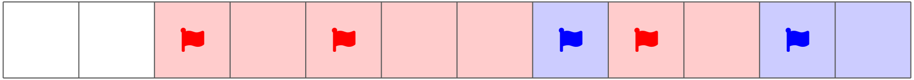
Problem Idea by `mtyeung1`

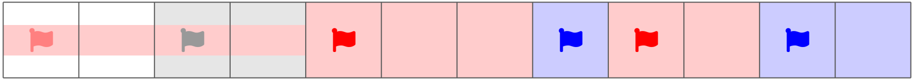
Preparation by `QwertyPi` (Thanks!)



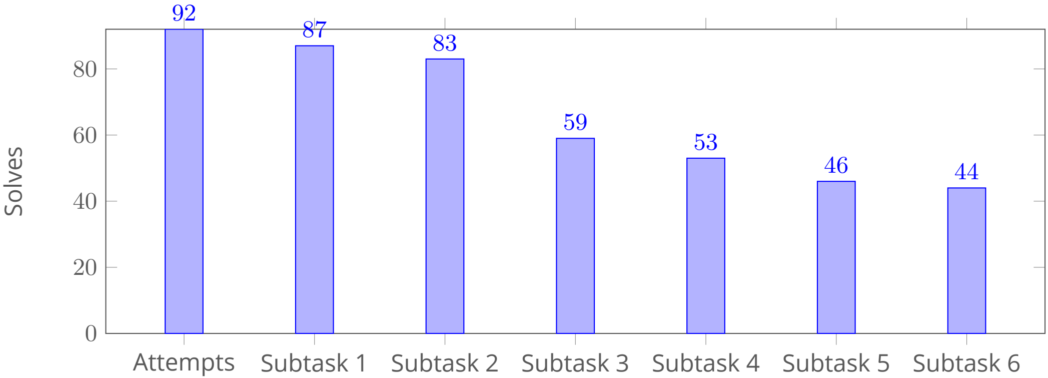
Problem Restatement

- Given a long line of M cells with N flags (either red or blue). A flag occupies the cells to the right before the next flag / the end of the line is reached.
- Move **at most one** red flag to maximize the red tribe's territory.





Statistics



First solved by **WYK23F32** (Xu Adam) at **9m 2s**.

Subtasks

For all cases: $1 \leq N \leq 10^5$, $1 \leq M \leq 10^9$, $1 \leq P_1 < P_2 < \cdots < P_N \leq M$.

Subtask	Points	Constraints
1	11	$N = 2$, $C_1 = \texttt{R}$, $C_2 = \texttt{B}$
2	12	$N \geq 2$, $C_1 = \texttt{R}$, $C_2 = \cdots = C_N = \texttt{B}$
3	17	$N \geq 2$, $C_1 = \texttt{R}$, $C_2 = \texttt{R}$
4	22	$N \leq 100$, $M \leq 500$
5	23	$N \leq 2000$
6	15	No additional constraints

Table of Contents

1 The Problem

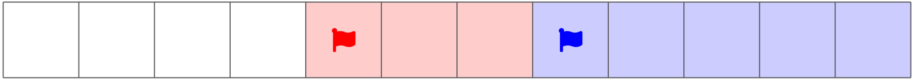
2 Subtask 1-3: Building Observations

3 Subtask 4-5: Efficient Exhaustions

4 Full Solution

Subtask 1 (11%): $N = 2$, $C_1 = \texttt{R}$, $C_2 = \texttt{B}$

- In this subtask, there are only two flags and we can only move the red flag:



- What is the optimal way to move the red flag?

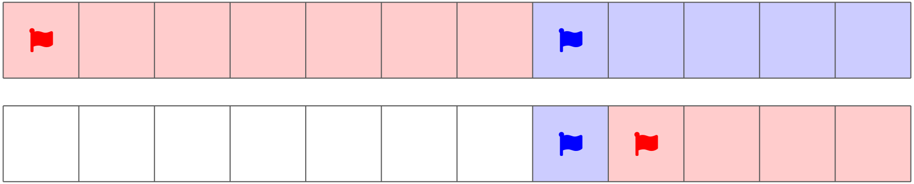
Subtask 1 (11%): $N = 2$, $C_1 = \texttt{R}$, $C_2 = \texttt{B}$

Observation 1

To **maximize** Red tribe's territory, the red flag can only end up at:

- The leftmost cell (cell 1).
- The cell right next to a blue flag (cell $P_i + 1$).

This ensures the territory cannot be expanded for free!



Subtask 1 (11%): $N = 2$, $C_1 = \texttt{R}$, $C_2 = \texttt{B}$

- In this subtask, the input is given in the format:

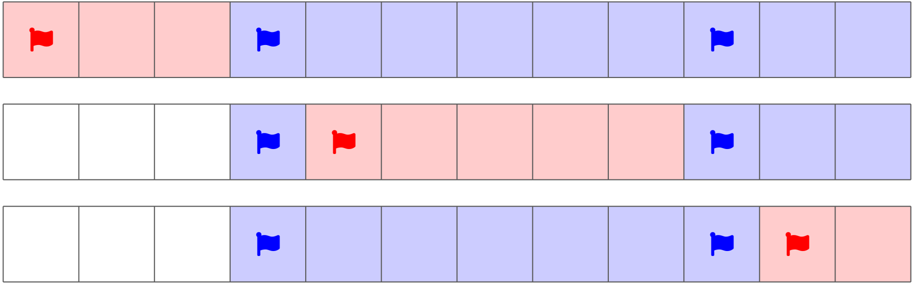
$\texttt{2}\ M$
$\texttt{R}\ P_1$
$\texttt{B}\ P_2$

- There are only two choices:
  - Occupying the leftmost cell: Total territory $= P_2 - 1$
  - Occupying the cell right next to P_2: Total territory $= M - P_2$

We can output the larger one!

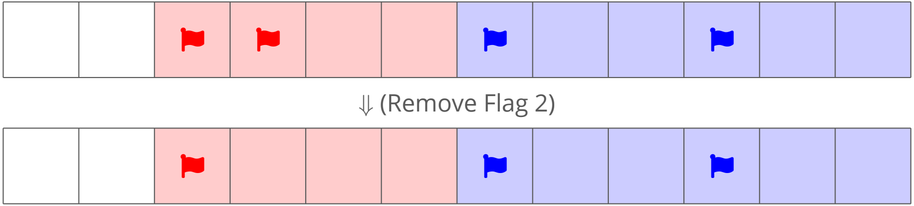
Subtask 2 (12%): $N \geq 2$, $C_1 = \texttt{R}$, $C_2 = \cdots = C_N = \texttt{B}$

- This subtask can be solved similarly – but there are more blue flags!
  - Occupying the leftmost cell: Total territory $= P_2 - 1$
  - Occupying the cell right next to P_i $(2 \leq i < N)$: Total territory $= P_{i+1} - P_i - 1$.
  - Occupying the cell right next to P_N: Total territory $= M - P_N$



Subtask 3 (17%): $N \geq 2$, $C_1 = \texttt{R}$, $C_2 = \texttt{R}$

- From this subtask onwards, note that we have to make **two decisions**: Decide which red flag to move and decide its final location. Subtasks 1-2 gave us a brief idea on where the flag should be placed.
- How does the constraint $C_1 = \texttt{R}, C_2 = \texttt{R}$ help us in picking a flag?



- Note that Flag 2 can be removed without affecting the territory!

Subtask 3 (17%): $N \geq 2$, $C_1 = \texttt{R}$, $C_2 = \texttt{R}$

> **Observation 2**
>
> Suppose there are two **consecutive** red flags. There exists an optimal solution where the second red flag is moved somewhere else.

Intuition.

- When we remove a flag, we only lose existing territory.
- When we insert a flag, we only gain new territory.
- Since removing the second red flag does not cause any loss in territory, removing this flag is always no worse than any other flags.

Subtask 3 (17%): $N \geq 2$, $C_1 = \texttt{R}$, $C_2 = \texttt{R}$

- We can first remove the second flag, and calculate the total territory of the red tribe.
  - For every red flag, we get a territory of $P_{i+1} - P_i$ (or $M - P_i + 1$ for the last flag).
- Next, we **insert** the removed flag, to maximize the amount of **new territory** we get.
  - Placing at the leftmost cell: New territory $= P_1 - 1$.
  - Placing right next to a **blue flag** P_i: New territory $= P_{i+1} - P_i - 1$ (or $M - P_N$).
- The final answer will be the total territory $+$ the maximum amount of new territory.

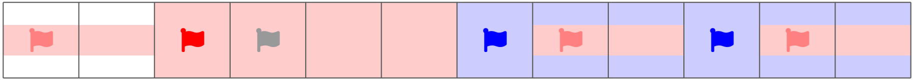


Table of Contents

1. The Problem
2. Subtask 1-3: Building Observations
3. Subtask 4-5: Efficient Exhaustions
4. Full Solution

Subtask 4 (22%): $N \leq 100$, $M \leq 500$

- The small constraints allow us to **brute force** over all possibilities.

- We can use the following algorithm:

```
for $i$ from $1$ to $N$ do                                                    ▷ Flag to remove, $O(N)$
    for $j$ from $1$ to $M$ do                                                ▷ Position to insert, $O(M)$
        Calculate the total territory if flag $i$ is moved to location $j$   ▷ $O(N)$
        if total territory is maximum so far then
            Update the answer
```

- The total time complexity is $O(N^2M)$.

Subtask 5 (23%): $N \leq 2000$

- In this subtask, the inputs are much larger so some optimizations are needed.
- **First Issue**: M can be as large as 10^9, so we should no longer brute force the position where the flag is inserted.
  - Based on **Observation 1**, we can only consider the leftmost cell and the cell right after the blue flags.
  - This optimizes the $O(M)$ loop to $O(N)$.
- **Second Issue**: It takes $O(N)$ time to calculate the total territory after the move, causing the total time complexity to be $O(N^3)$.
  - After deciding which flag to remove ($O(N)$ possibilities), we can follow Subtask 3's solution to find the maximum territory of the red tribe in $O(N)$.
  - This optimizes the total time complexity to $O(N^2)$.

Table of Contents

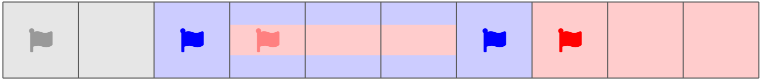
1. The Problem
2. Subtask 1-3: Building Observations
3. Subtask 4-5: Efficient Exhaustions
4. Full Solution

Full Solution

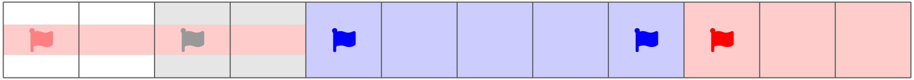
Observation 3

The decision of removing a flag and the decision of putting a flag is **MOSTLY** independent.

1. If we remove a flag and place it somewhere else ⇒ independent.
   - New territory = Cells gained by adding the flag − Cells lost by removing the flag.
   - We can first remove a flag that **minimizes** the number of cells lost, then add a flag that **maximizes** the number of cells gained.



2. If we move a flag slightly to the left ⇒ **dependent**!



Full Solution

We simply handle the two cases separately:

1. Remove a flag and place it somewhere else.
   - Pick a flag to remove that minimizes the number of cells lost. // $O(N)$
     Idea. For each flag, calculate the territory it occupies. Pick the smallest one.
   - Pick a position to add a flag that maximizes the number of cells gained. // $O(N)$
     Idea. Consider the leftmost position & all positions right next to a blue flag.
2. Move a flag slightly to the left.
   - For each red flag, **if the previous flag is a blue flag**, try moving it forward until the blue flag is reached. // $O(N)$
     New territory $= P_i - P_{i-1} - 1$.

Takeaways

- Subtasks guide you to think!
  - The full solution of a task usually requires several major **observations**.
  - Subtasks will guide you to unveil these observations, piecing together the solution.

- Verify your ideas carefully!
  - While removing/adding the flag seems to be two independent decisions, further thinking will reveal that they are not always independent.
  - It's important to try justifying your observations, and **corner cases** might show up along the way!

- Sample tests are not meant to be exhaustive (they are here to help you understand the task). It's important to **design your own test cases** while debugging.