現代的な機械 解説

米田 優峻 (E86120)

第22回情報オリンピック　本選5

0 問題概要

- N 個のライトタイルと M 個のボタンからなる「JOI マシン」があります
- ボタン i を押すと、次のことが起こります

1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、以下が繰り返される
   - 青色のタイルの上にボールがある場合：ボールが 1 つ左に移動し、タイルの色が赤になる
   - 赤色のタイルの上にボールがある場合：ボールが 1 つ右に移動し、タイルの色が青になる

- ボタン $L_i, L_i + 1, \ldots, R_i$ をその順に押したときの最終的な赤タイルの個数を求めよ、という質問が Q 個与えられるので、それぞれ答えてください

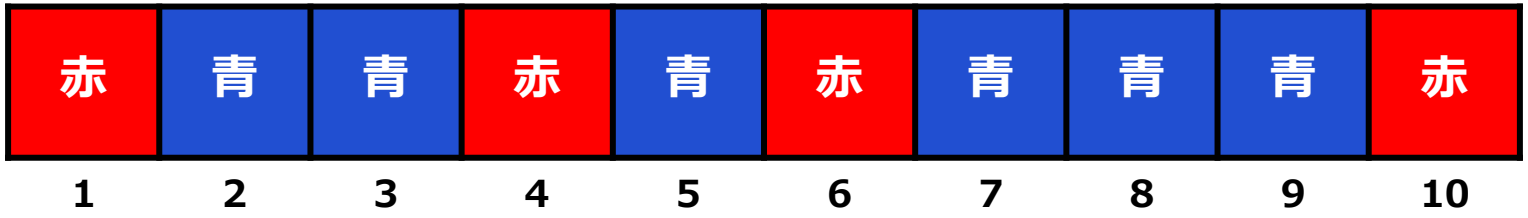
0 問題概要

問題設定が少し複雑なので、例を示します

0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

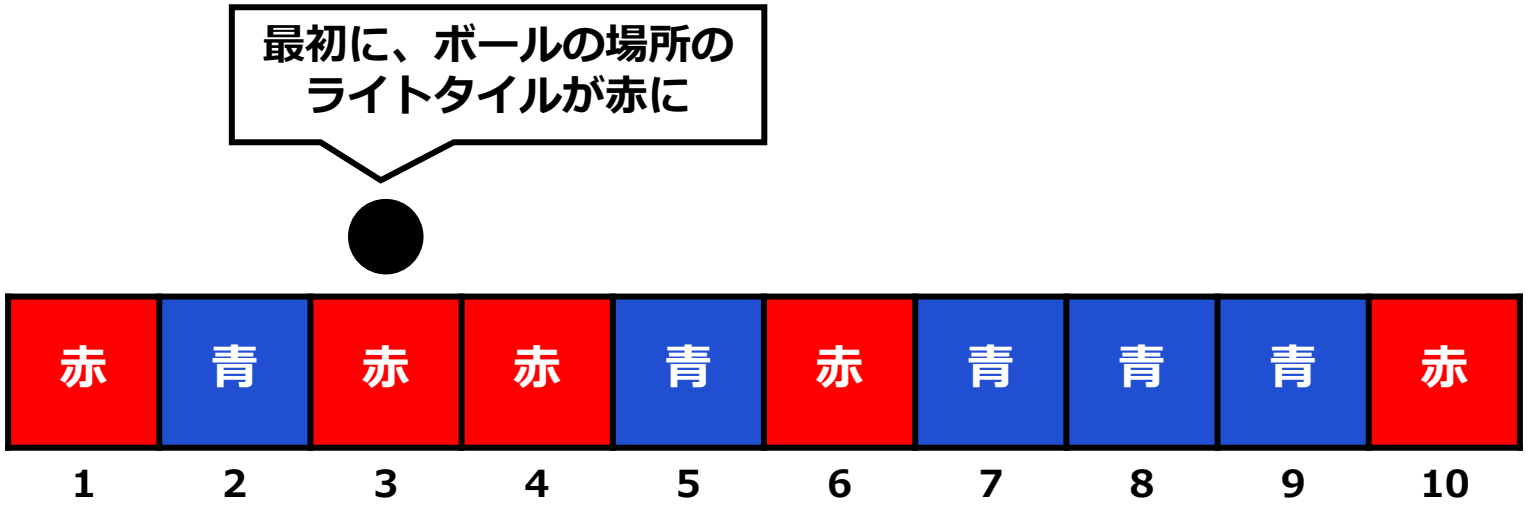
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

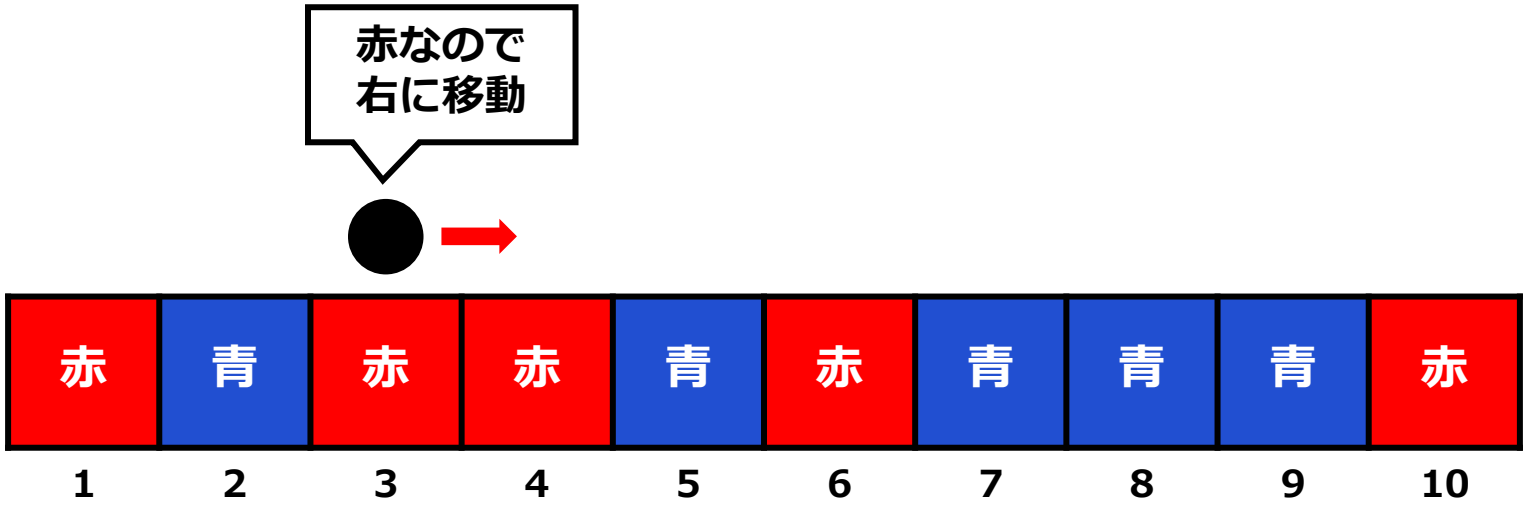
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

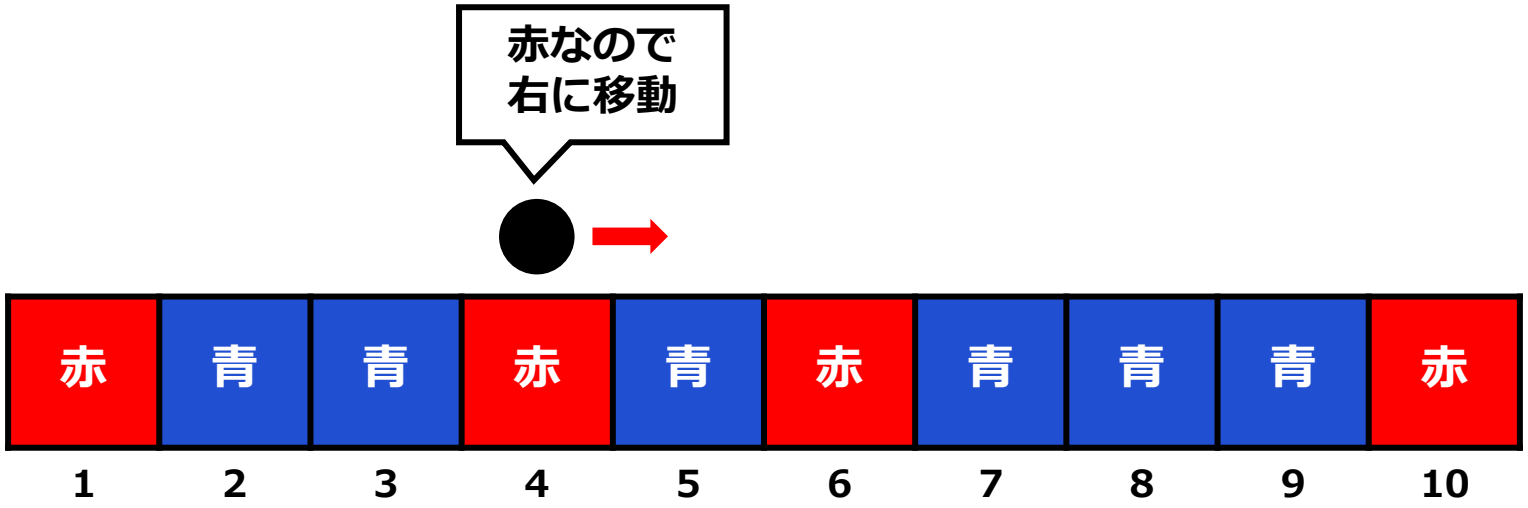
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

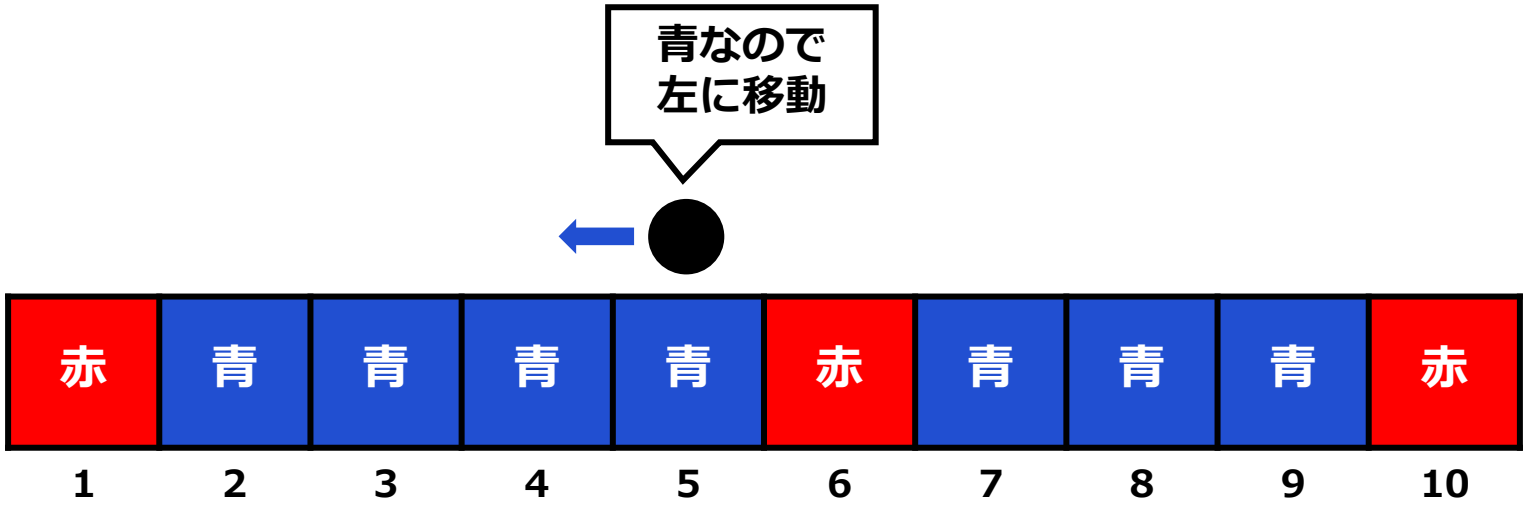
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

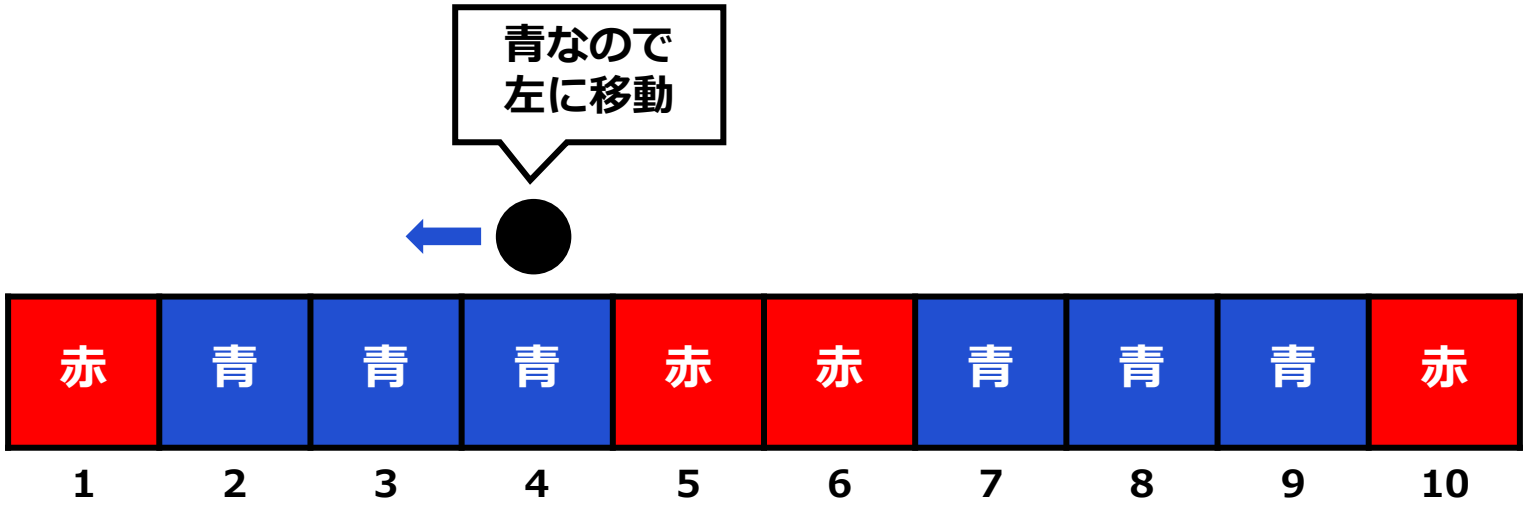
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

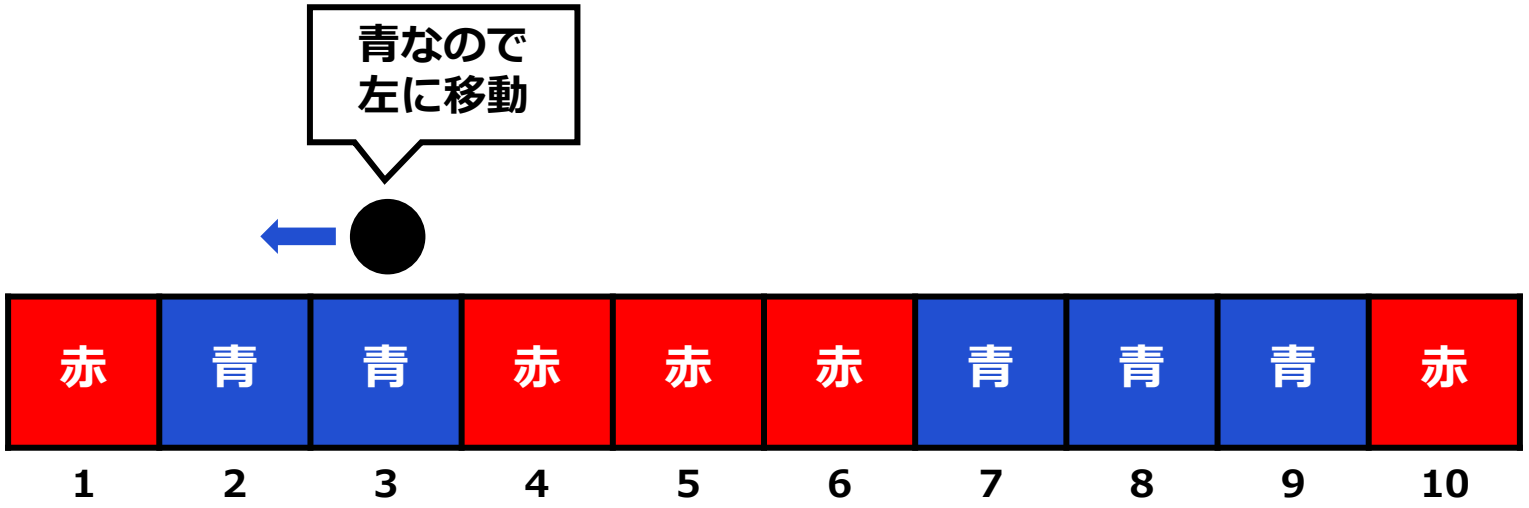
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

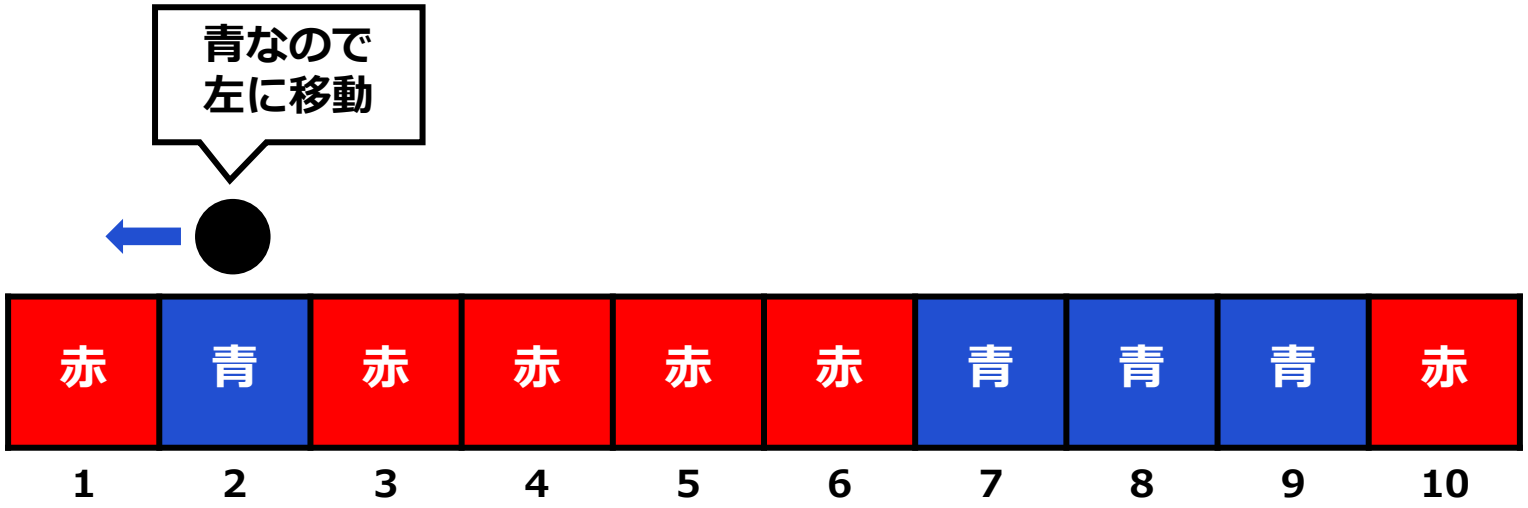
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

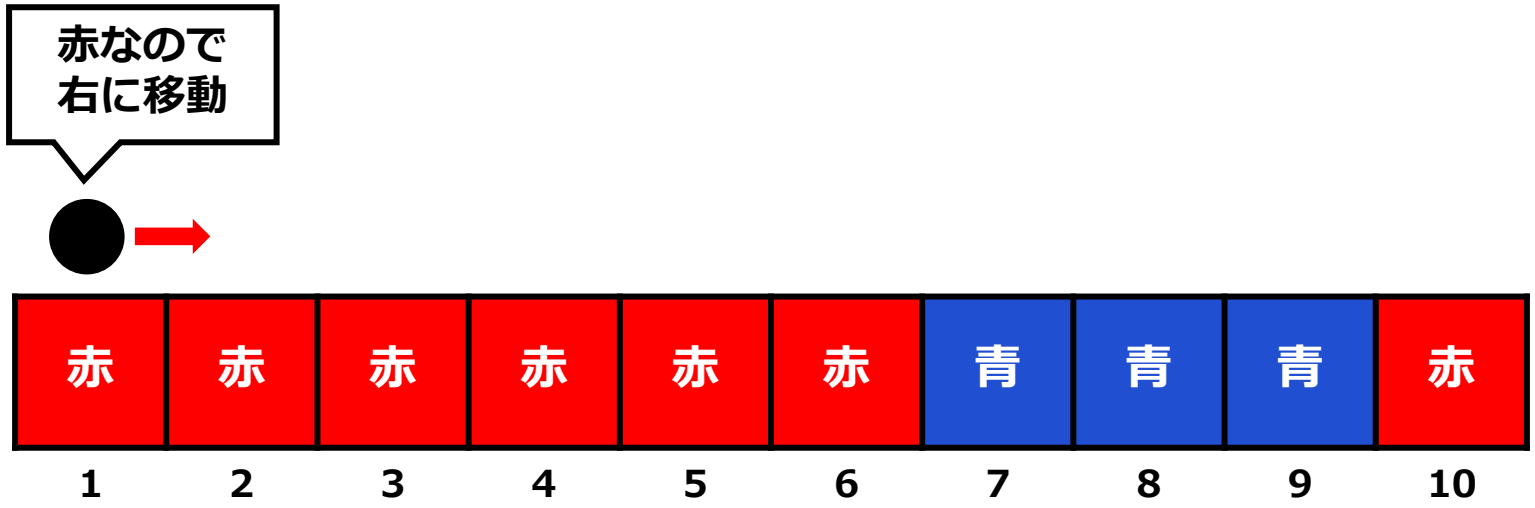
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

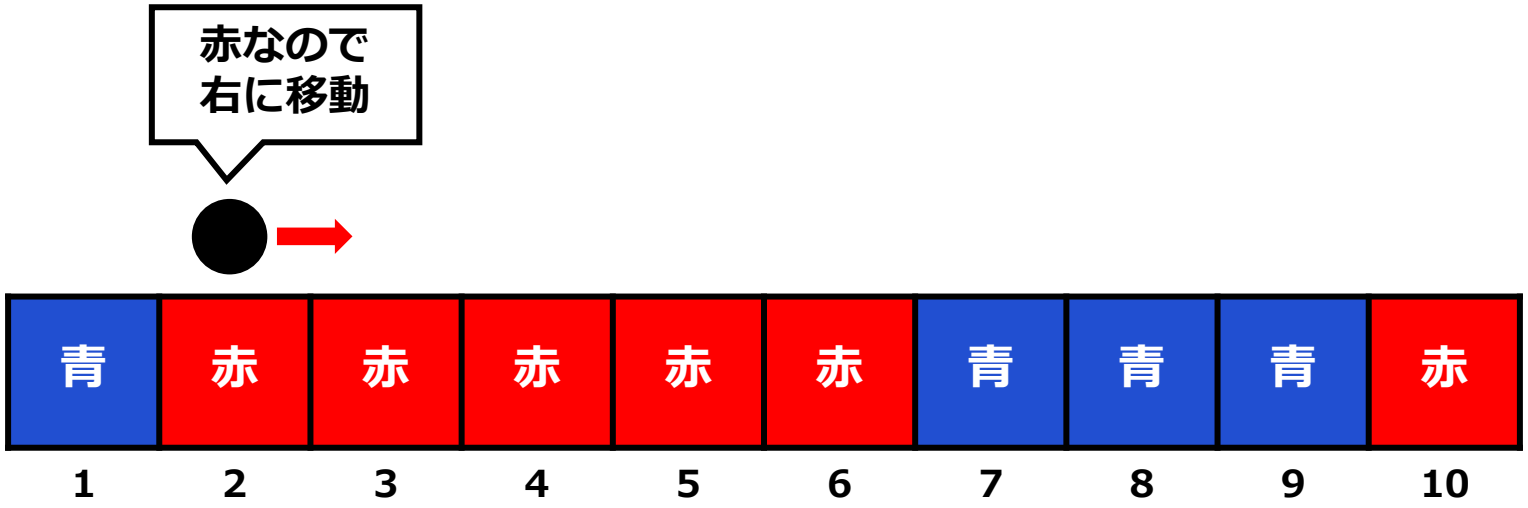
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

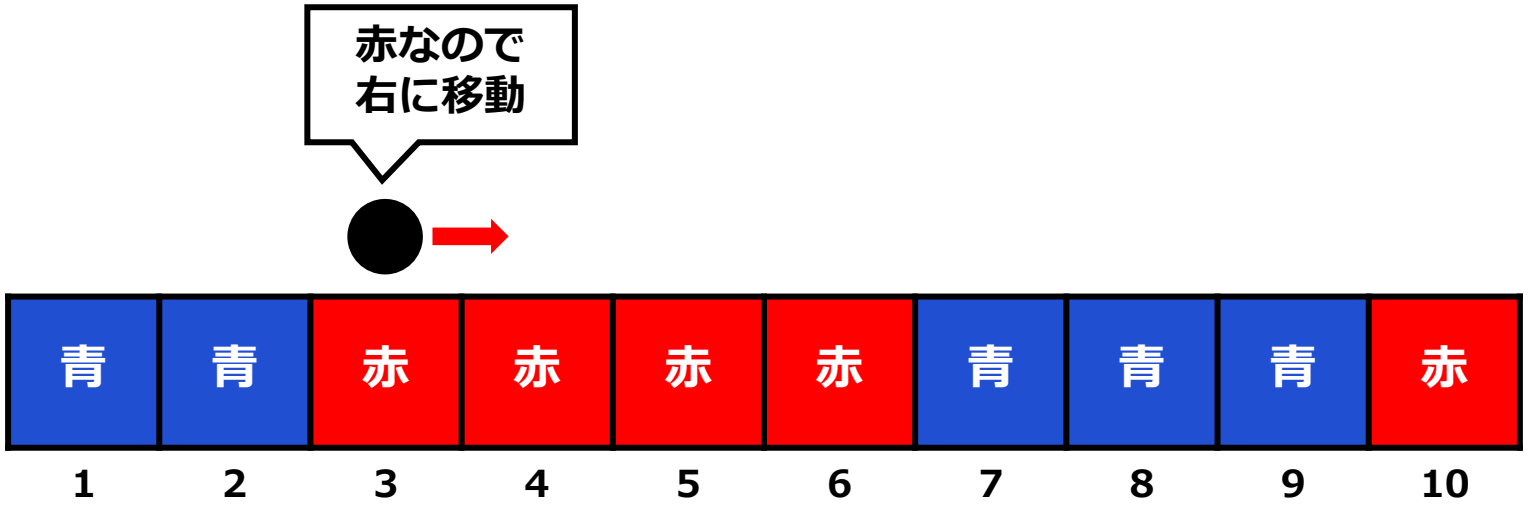
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

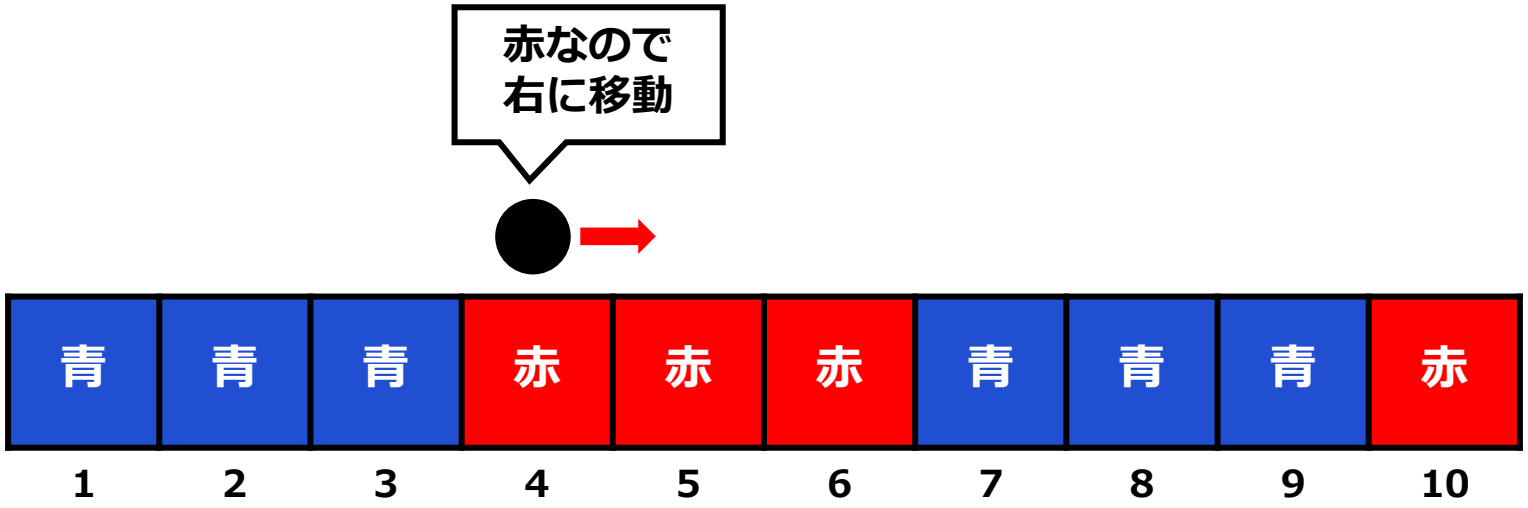
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

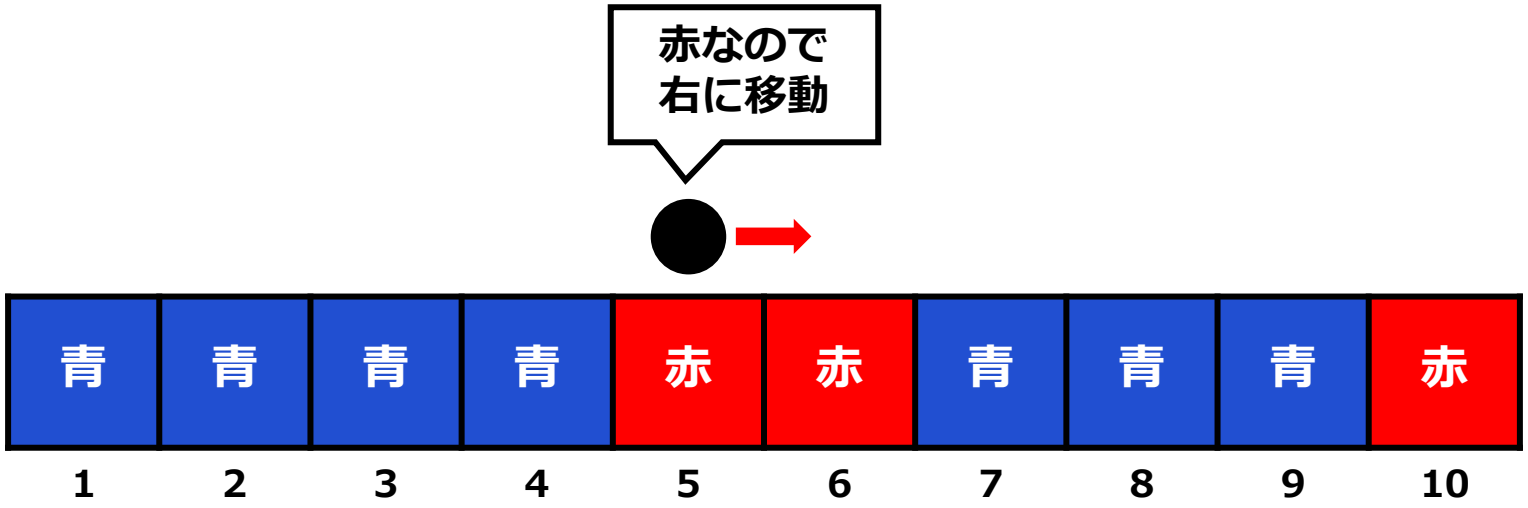
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

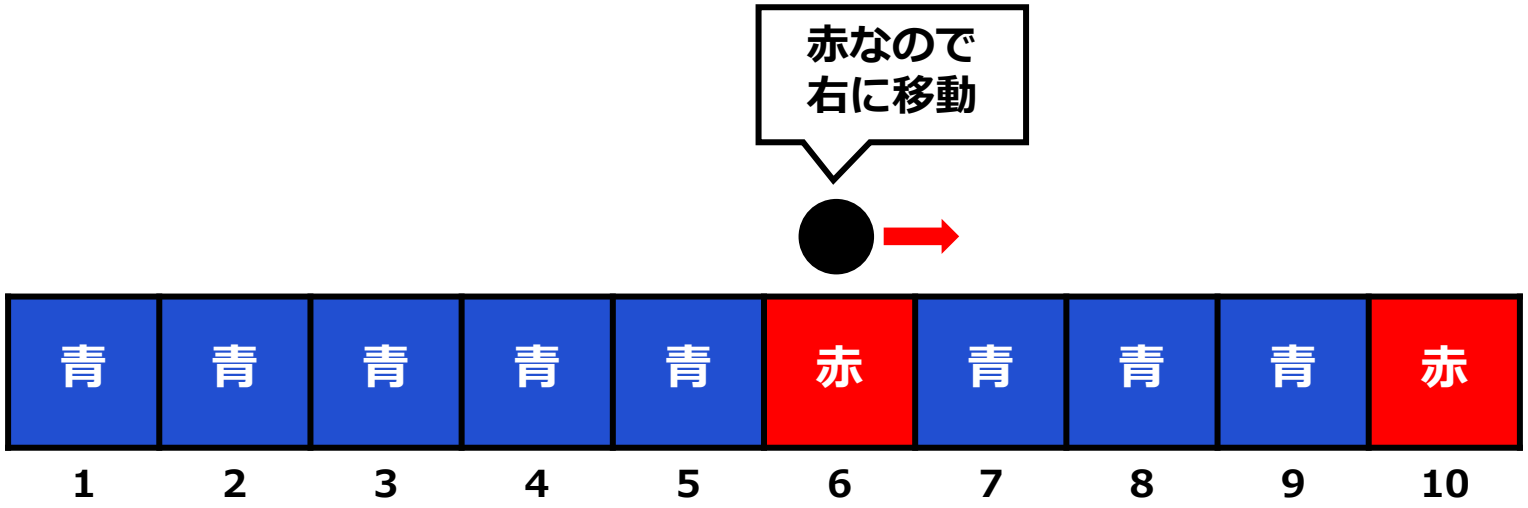
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

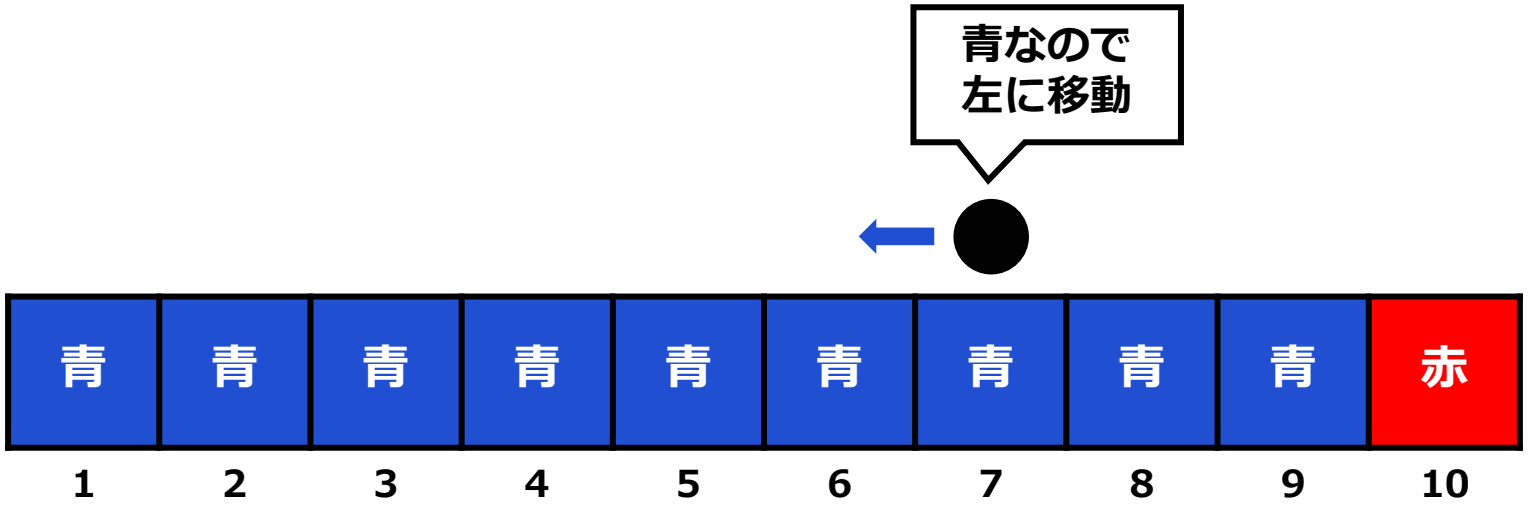
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

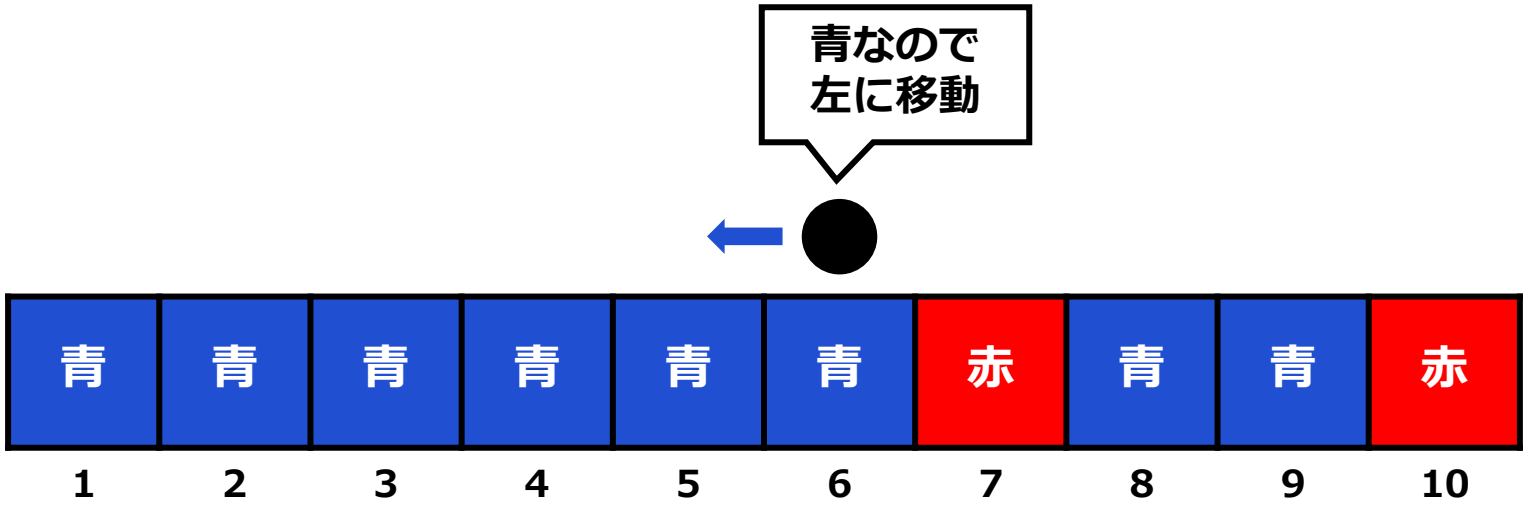
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

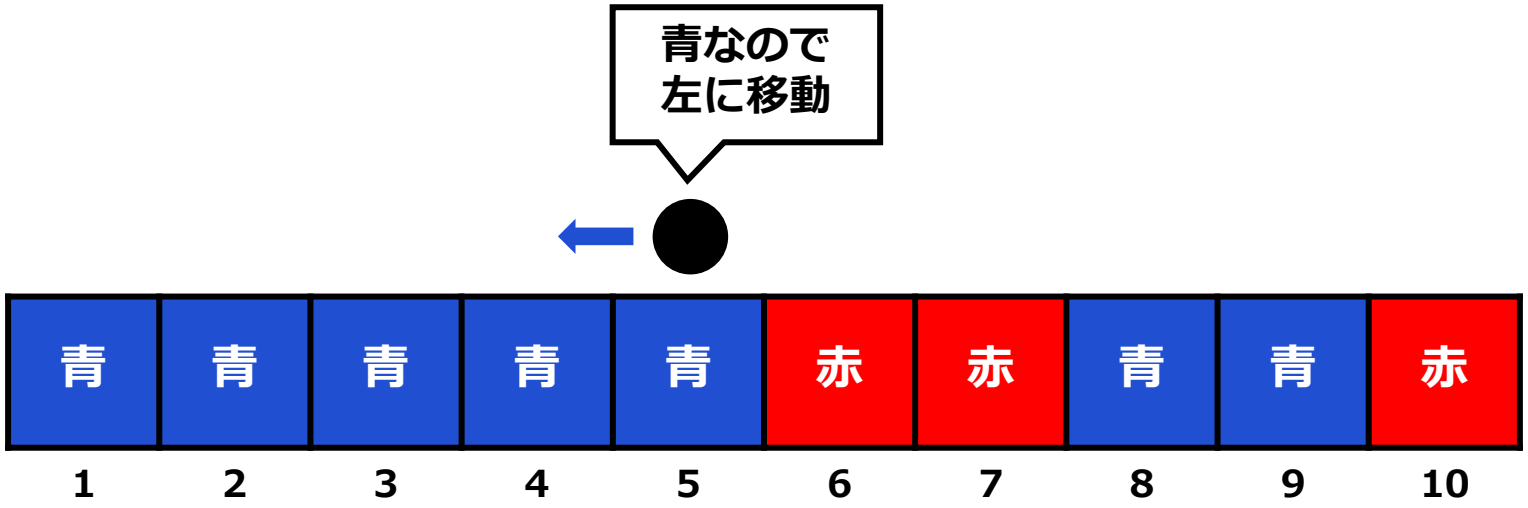
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

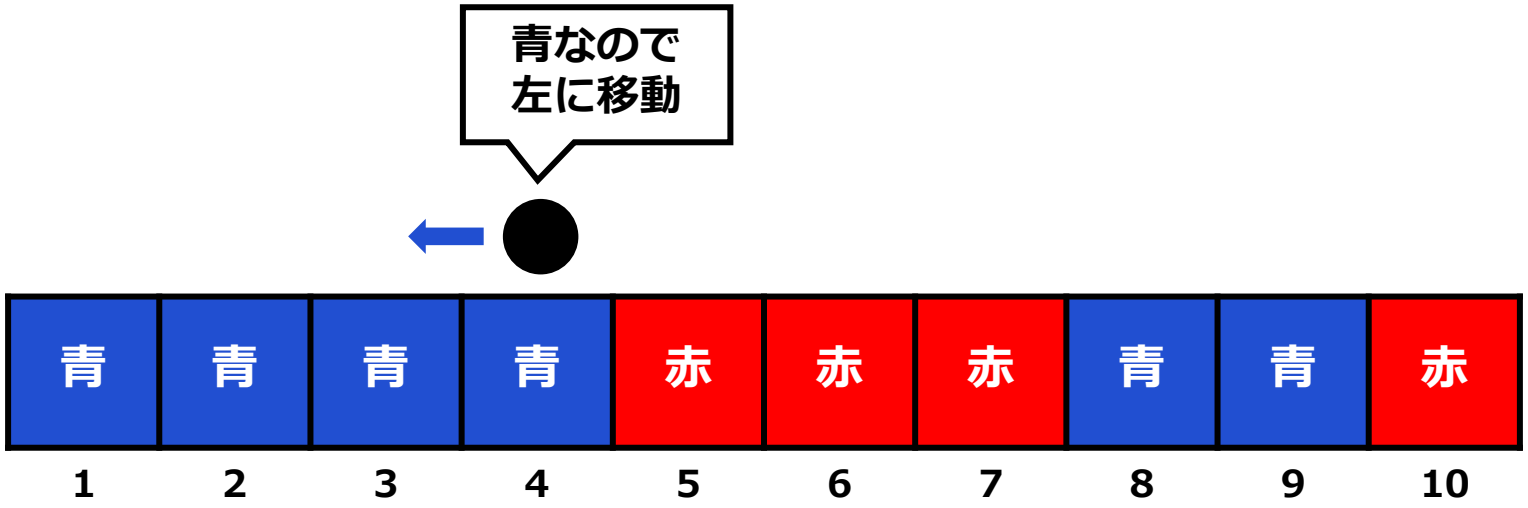
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

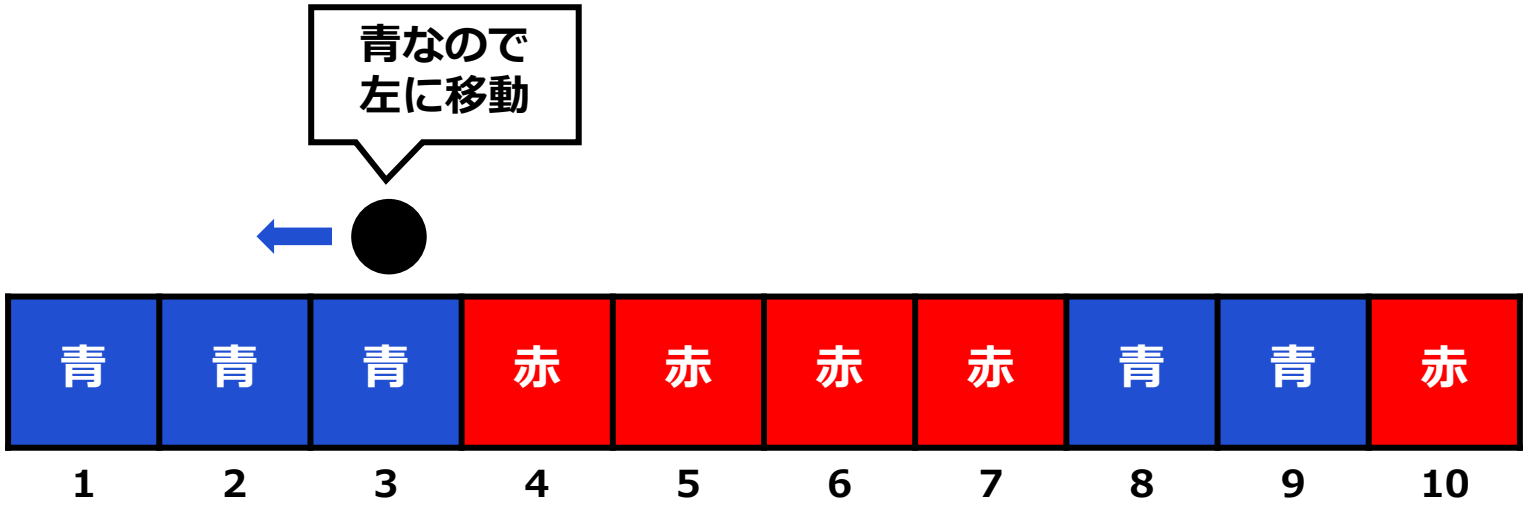
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

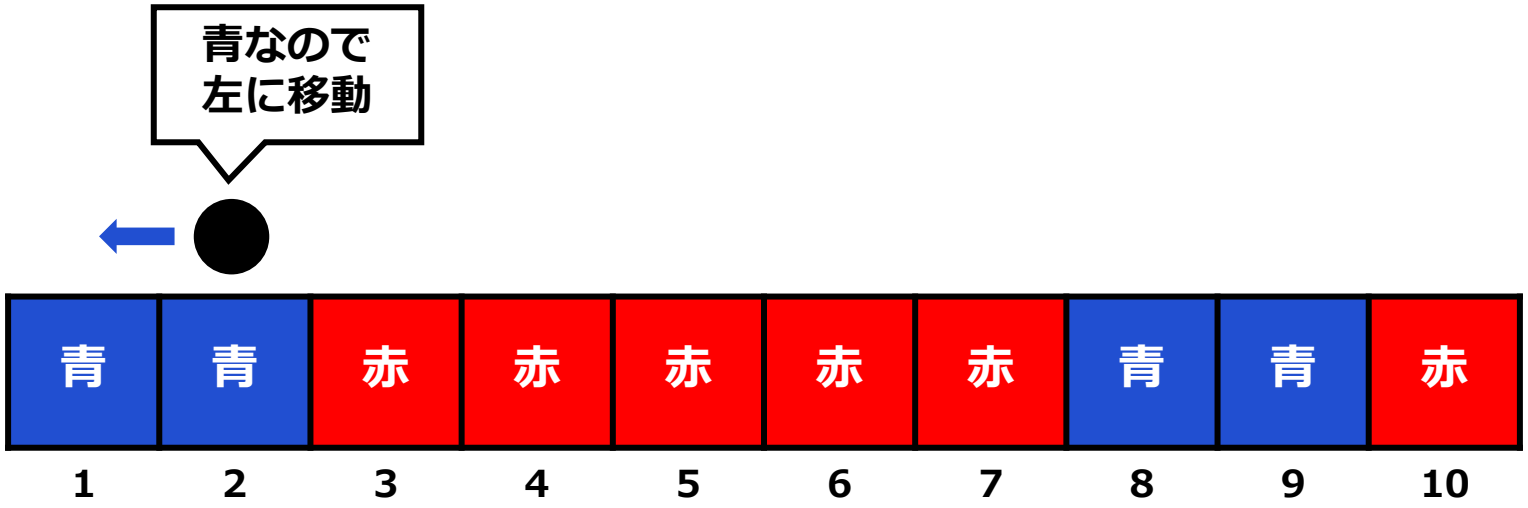
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

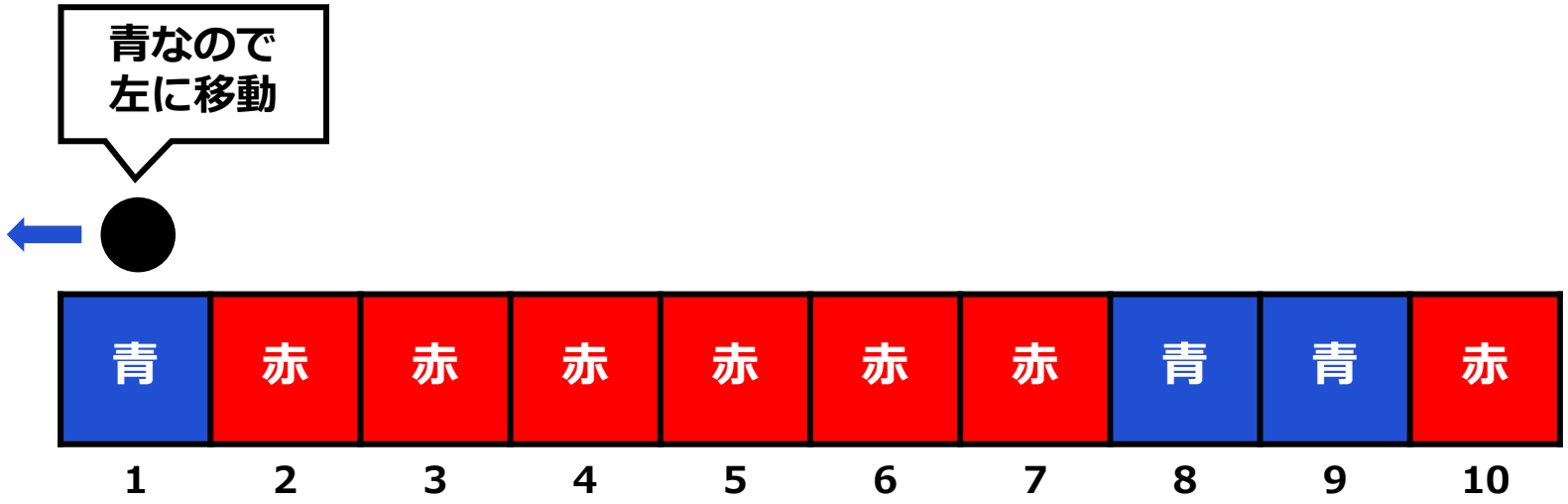
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

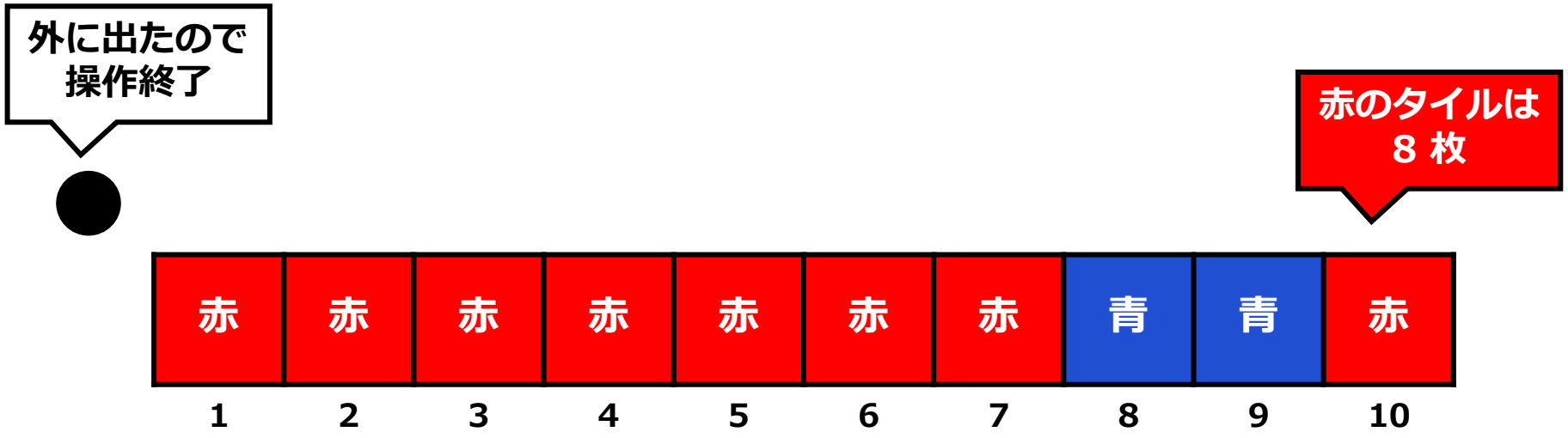
1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



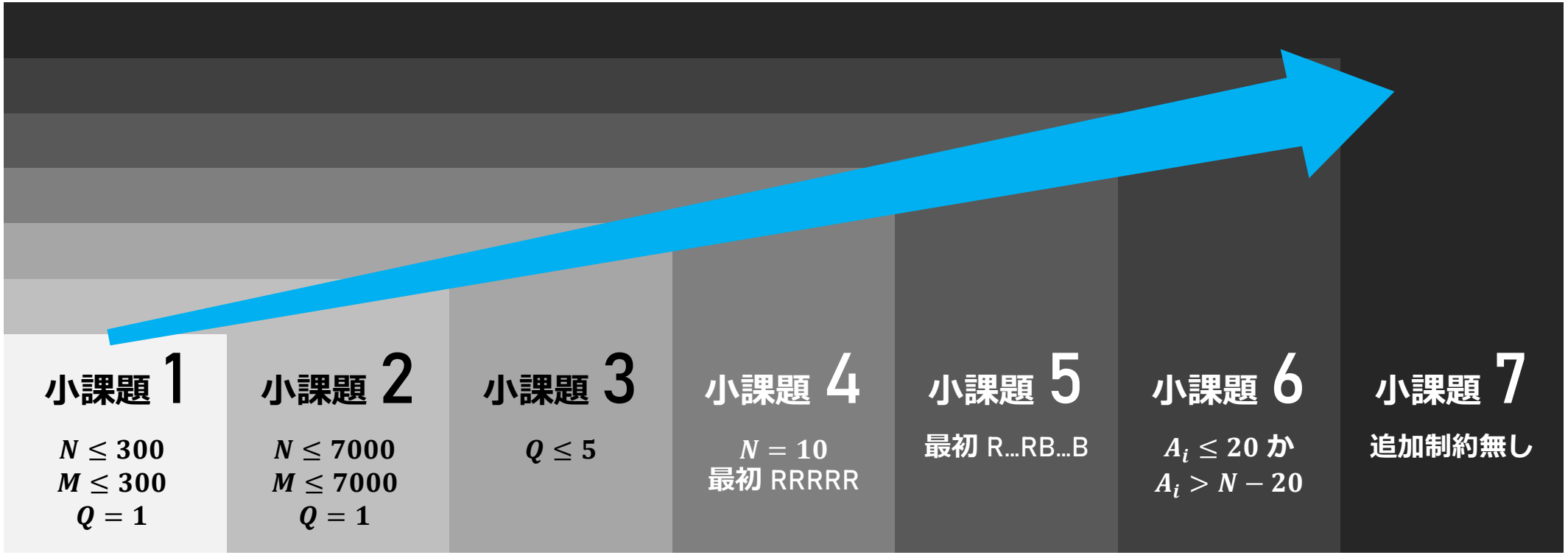
0 問題概要：例

例：初期状態が「赤青青赤青赤青青青赤」で、ボールをタイル 3 の上に置いた場合どうなるか？

1. ボールがライトタイル A_i の上に置かれ、ライトタイル A_i の色が赤色に変わる
2. ボールが外に出るまで、移動／塗り替えを繰り返す



0 満点へのロードマップ



※満点の制約は $N, M, Q \leq 120000$

小課題 1

$N \leq 300, M \leq 300, Q = 1$

1 小課題1（累計3点）

ボールの動きを直接シミュレーションすると、解ける

理由

- 1 回ボタンを押した後のステップ数は、$O(N^2)$ 回以下になるため
- ボタンは高々 M 回押されるため

➡ 計算量は $O(N^2M)$ なので、$N \leq 300, M \leq 300$ で間にあう

3点

1 小課題1（累計3点）

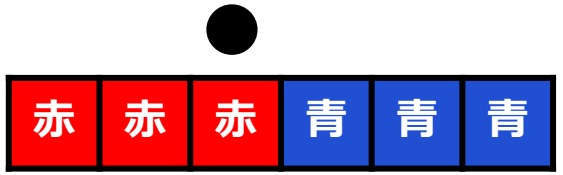
ボールの動きを直接シミュレーションすると、解ける

理由

- 1 回ボタンを押した後のステップ数は、$O(N^2)$ 回以下になるため
- ボタンは高々 M 回押されるため

➡ 計算量は O で間にあう

こういうケースで $\Theta(N^2)$ 回かかる



3点

小課題 2

$$N \leq 2000, M \leq 2000, Q = 1$$

2 重要な考察

ボールをタイル x の上に置いたときの、操作後の盤面は…

ボールが左から出た場合

左から x 番目の赤タイルまでが青になる

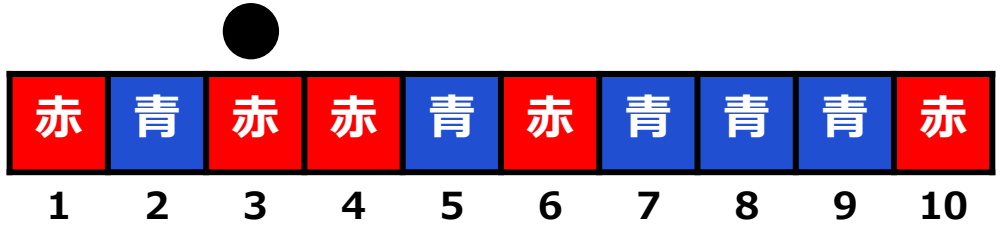
ボールが右から出た場合

右から $N+1-x$ 番目の青タイルまでが赤になる

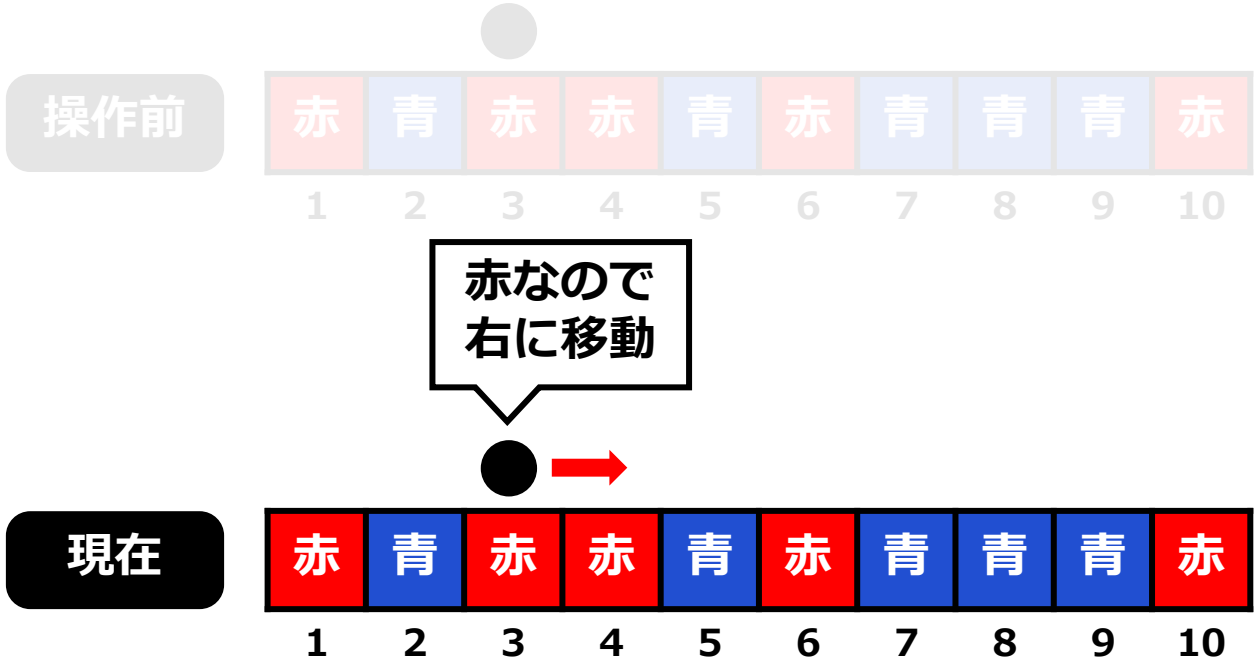
※「左から○○番目」「右から△△番目」というのは、タイル x が赤に変わった後の話

2 重要な考察：具体例

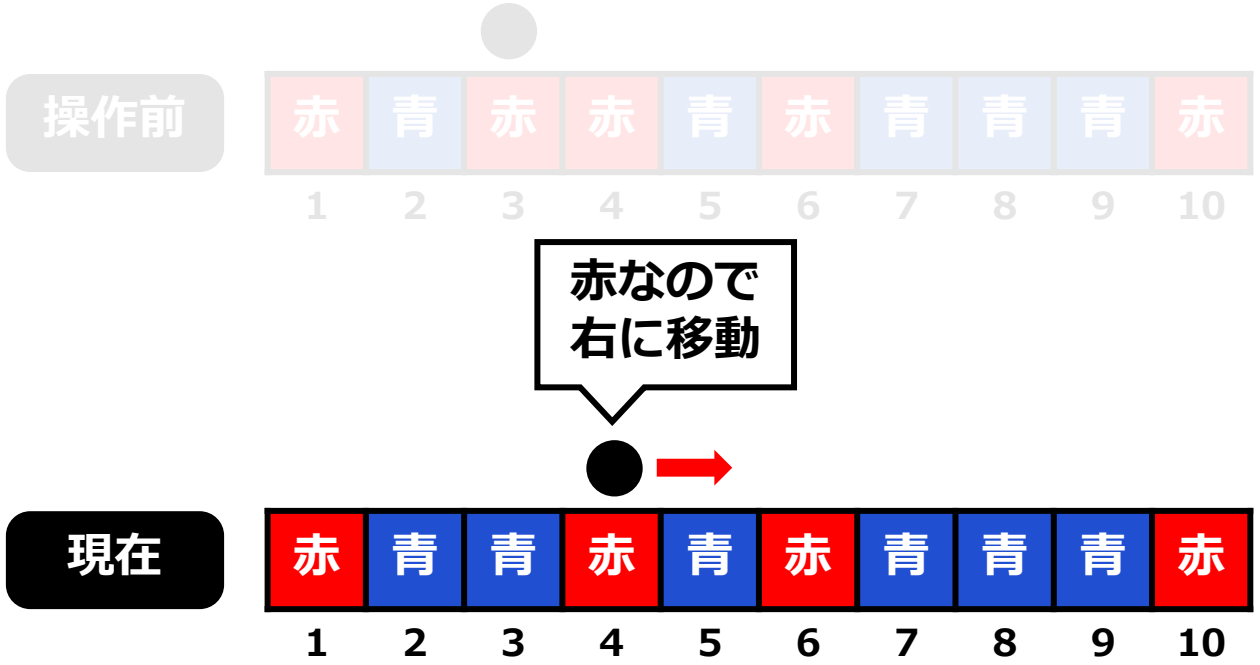
操作前



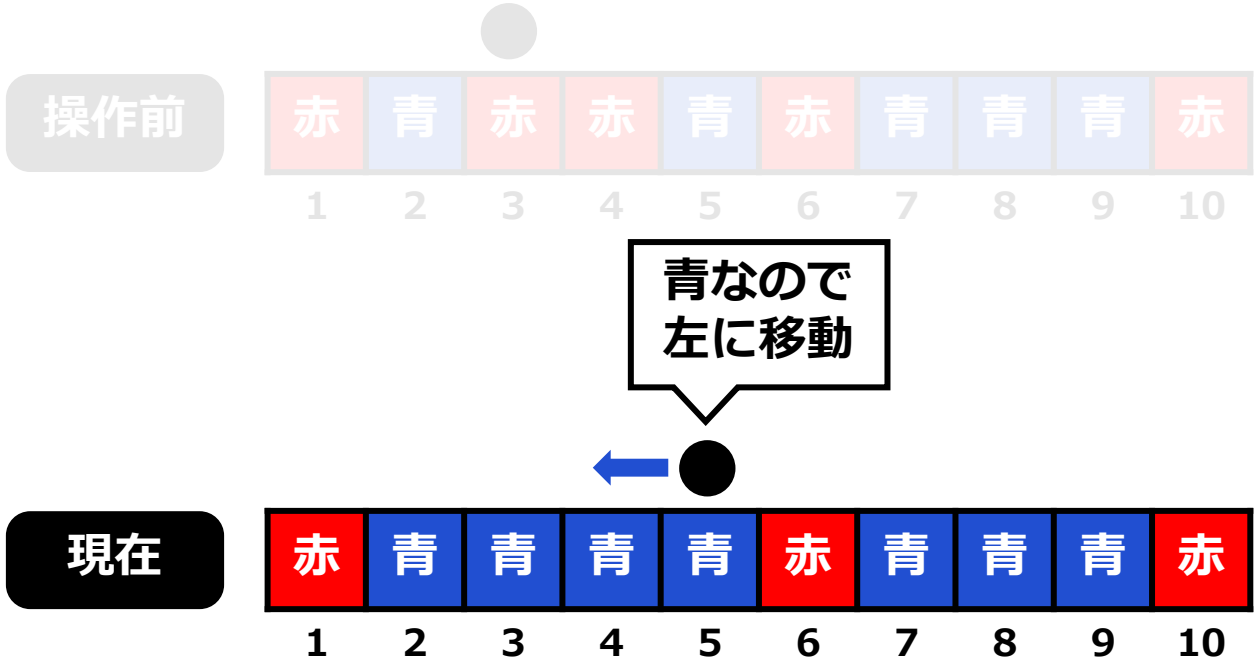
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
赤なので
右に移動
現在
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

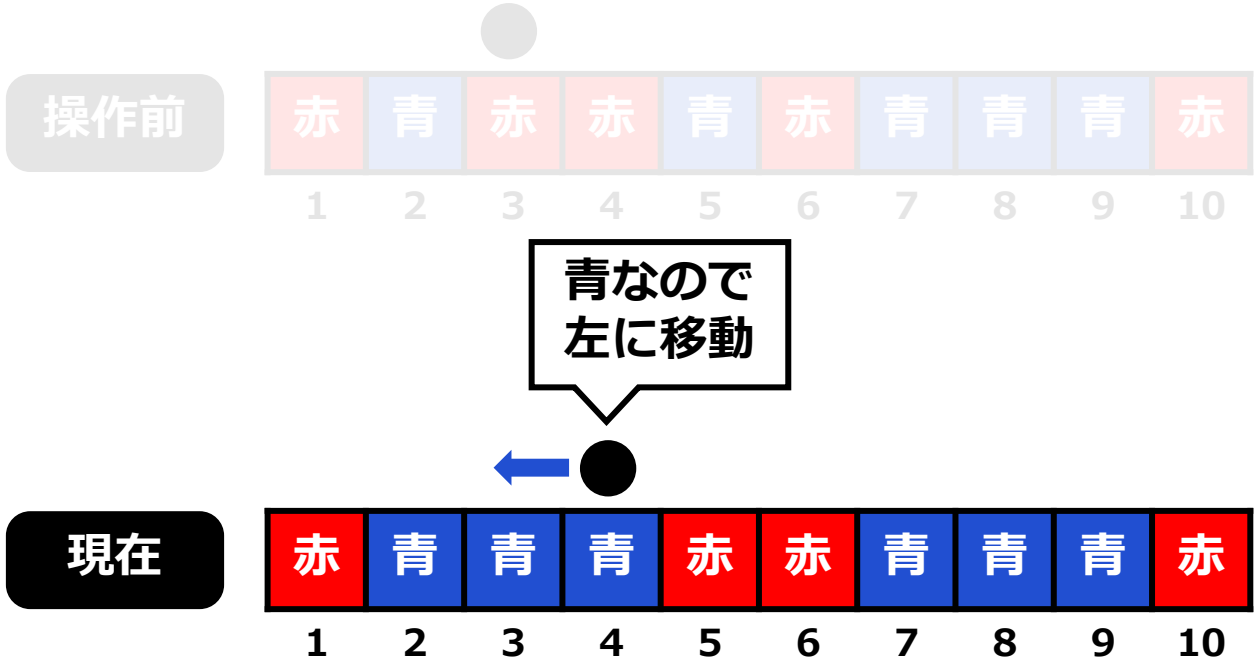
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
赤なので
右に移動
現在
赤 青 青 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

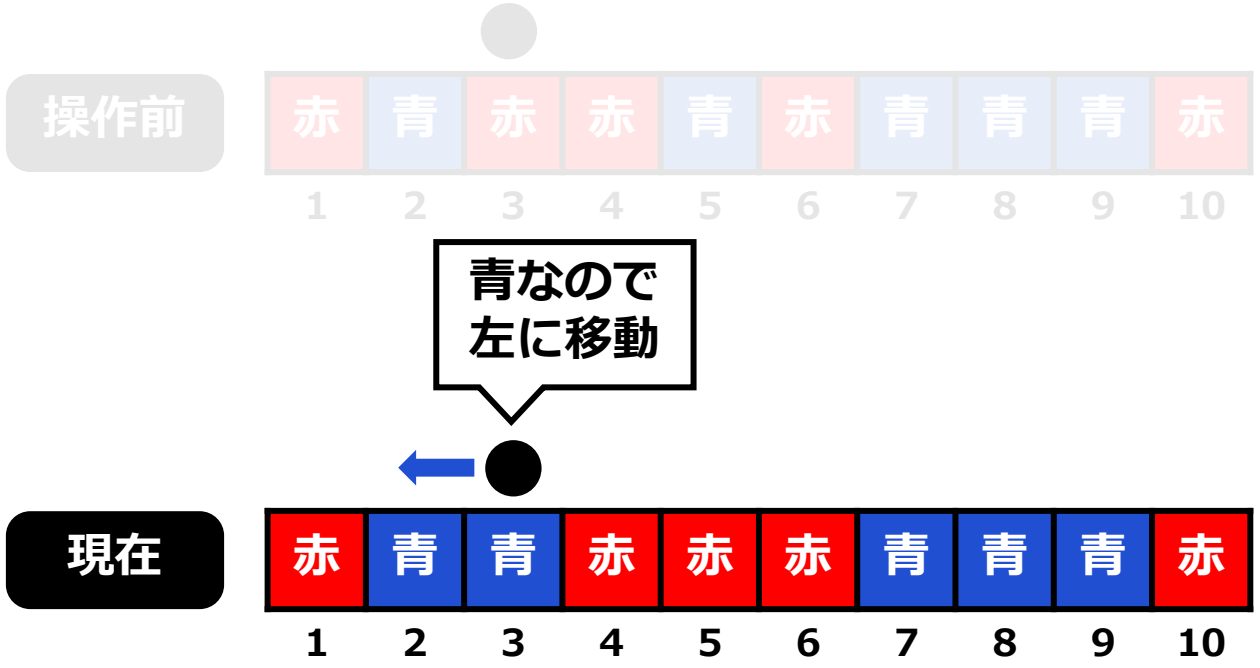
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
赤 青 青 青 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

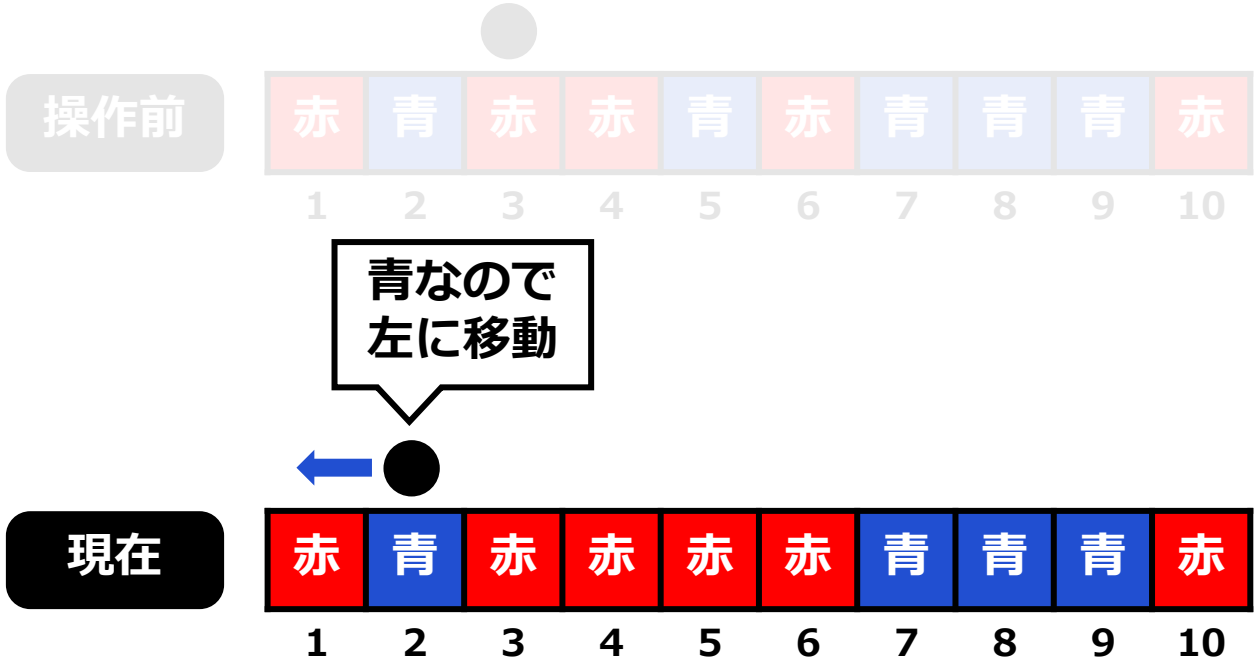
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
赤 青 青 青 赤 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

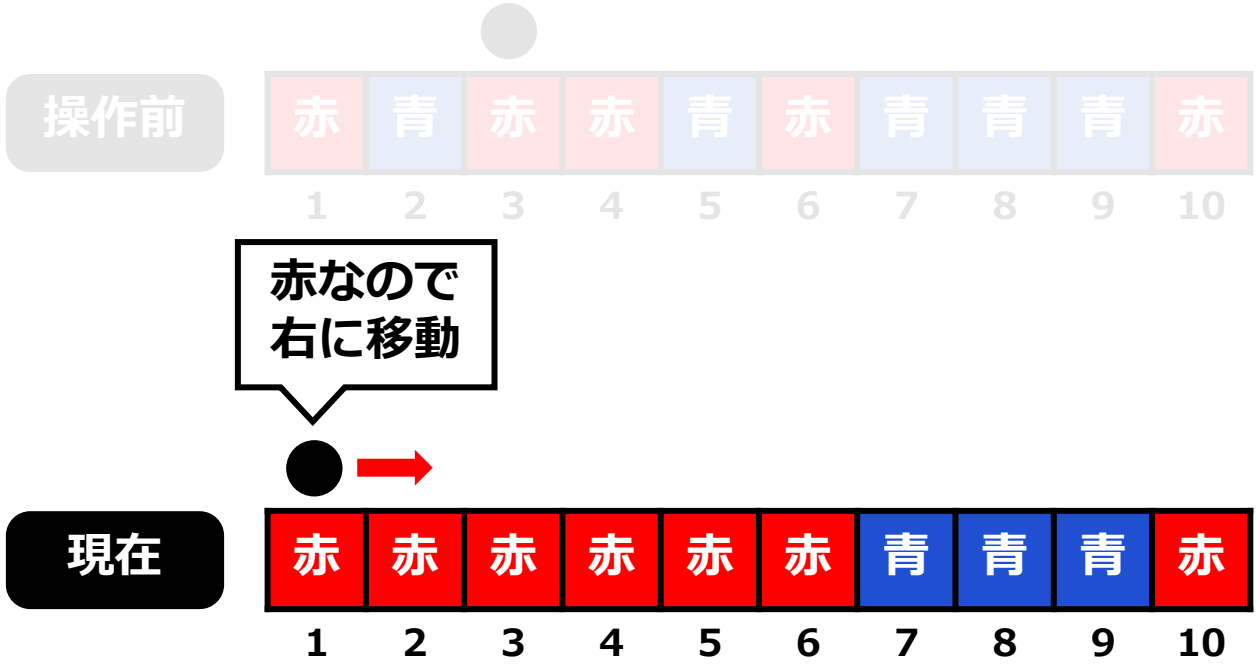
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
赤 青 青 赤 赤 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

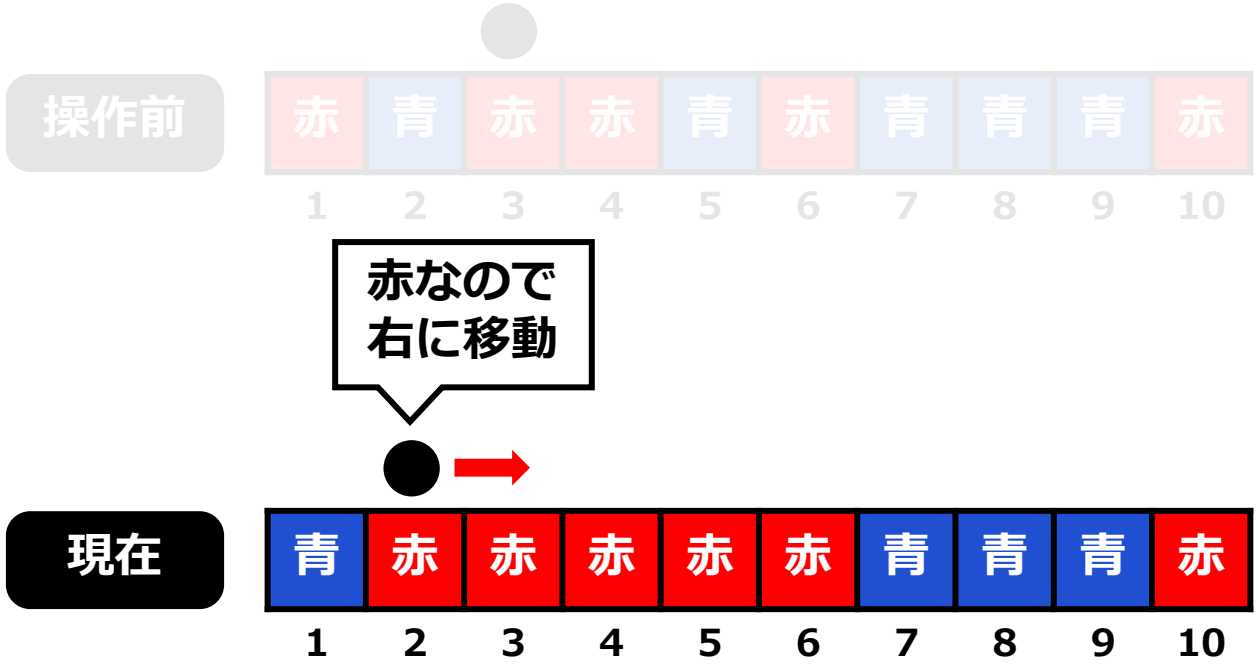
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
赤 青 赤 赤 赤 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

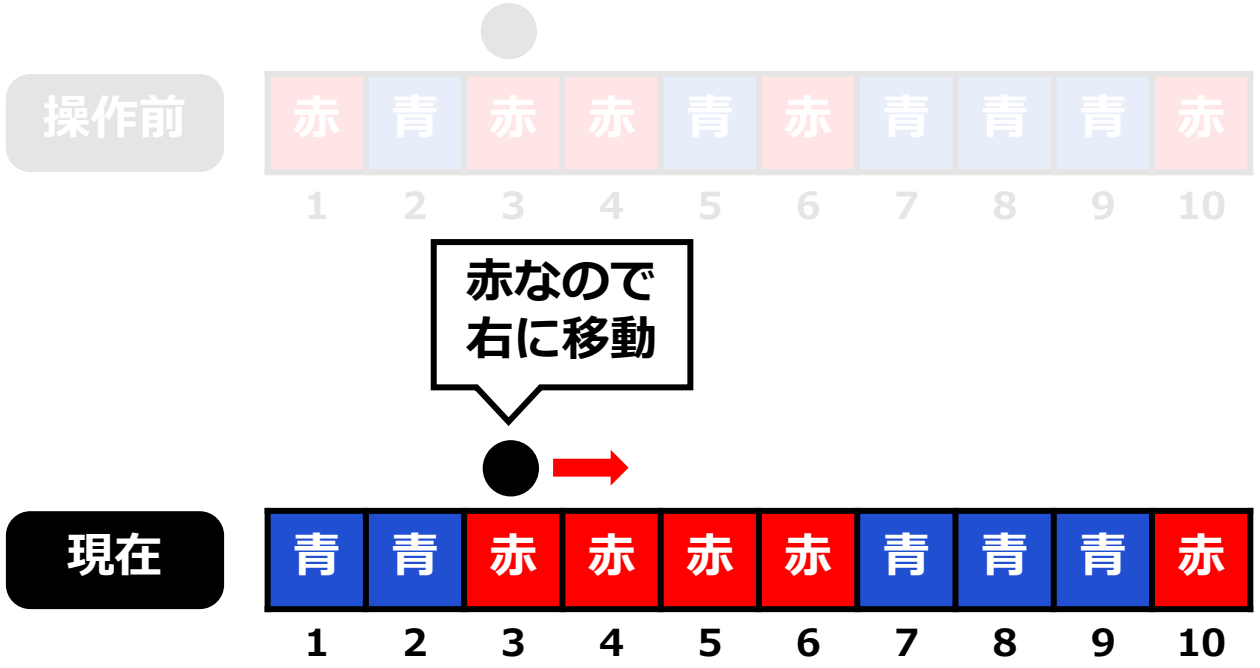
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
赤なので
右に移動
現在
赤 赤 赤 赤 赤 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

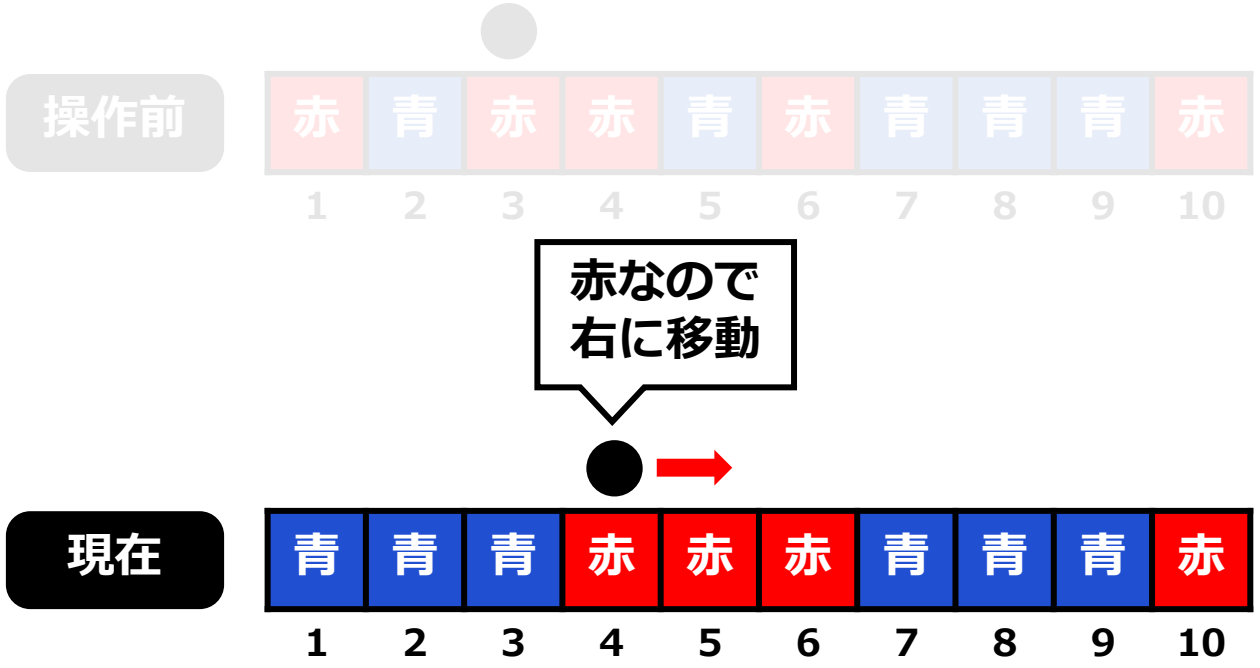
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
赤なので
右に移動
現在
青 赤 赤 赤 赤 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

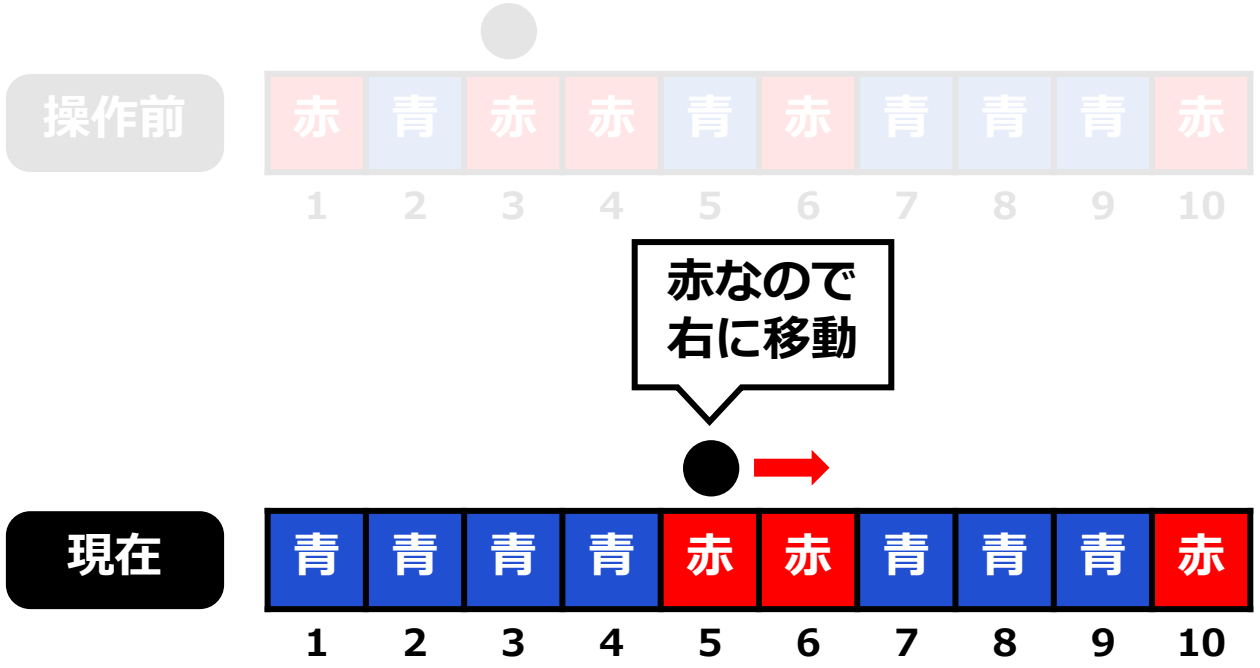
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
赤なので
右に移動
現在
青 青 赤 赤 赤 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

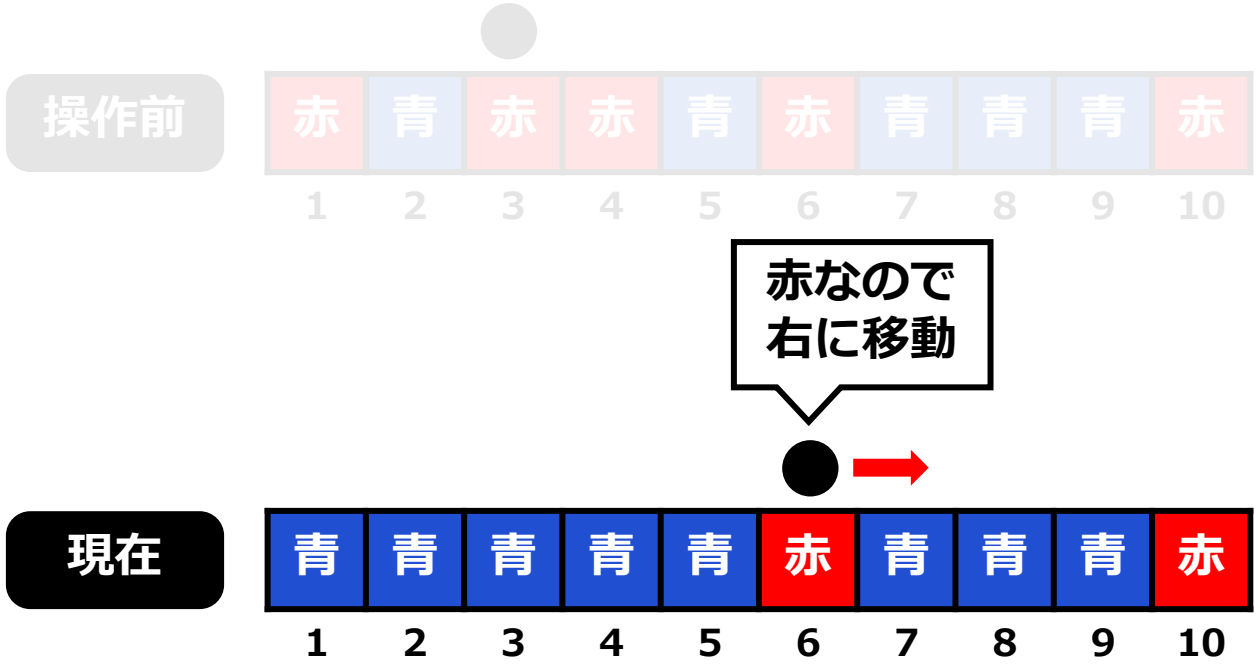
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
赤なので
右に移動
現在
青 青 青 赤 赤 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

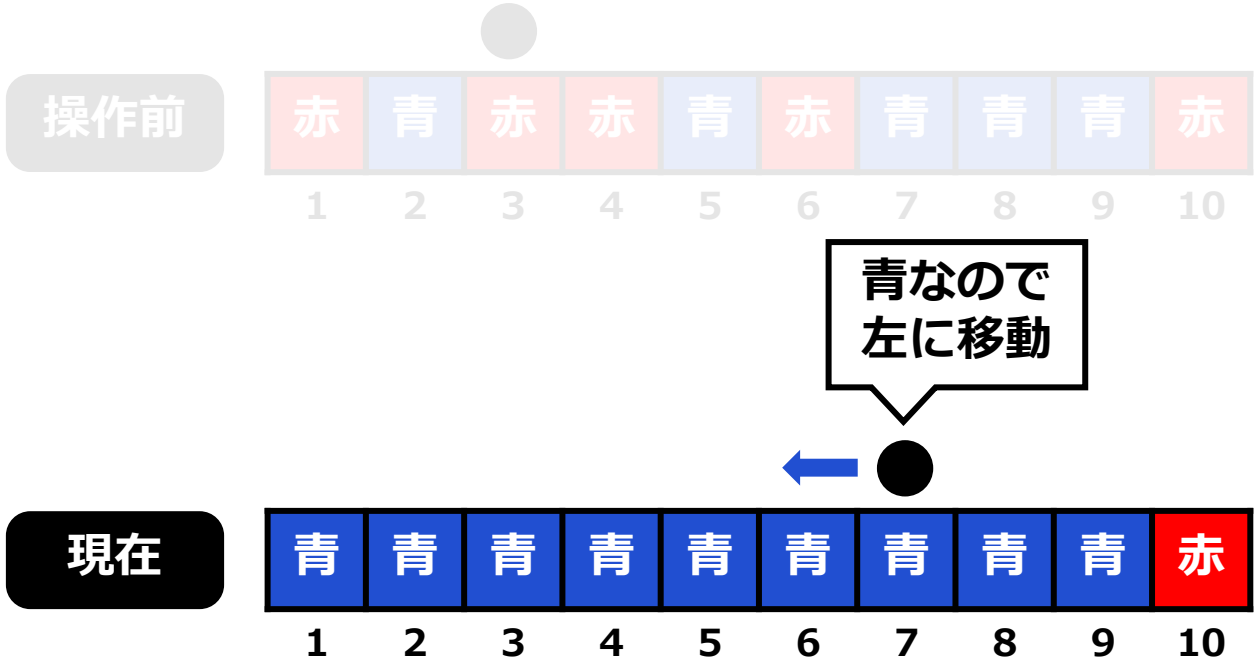
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
赤なので
右に移動
現在
青 青 青 青 赤 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

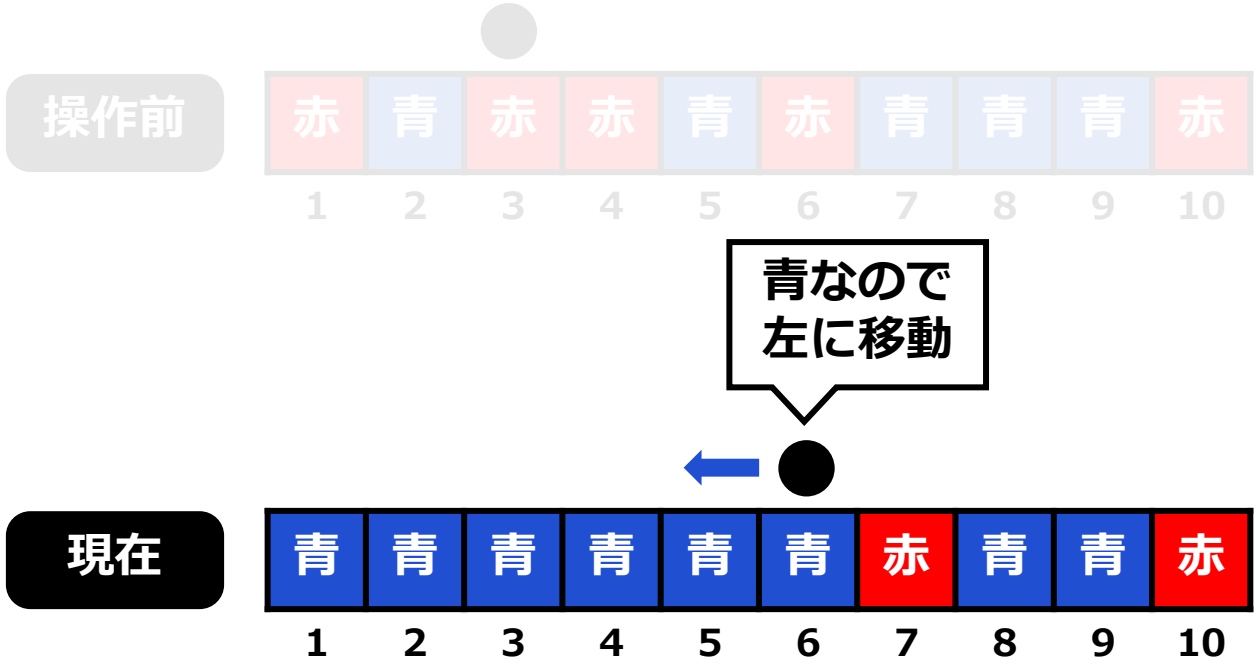
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
赤なので
右に移動
現在
青 青 青 青 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

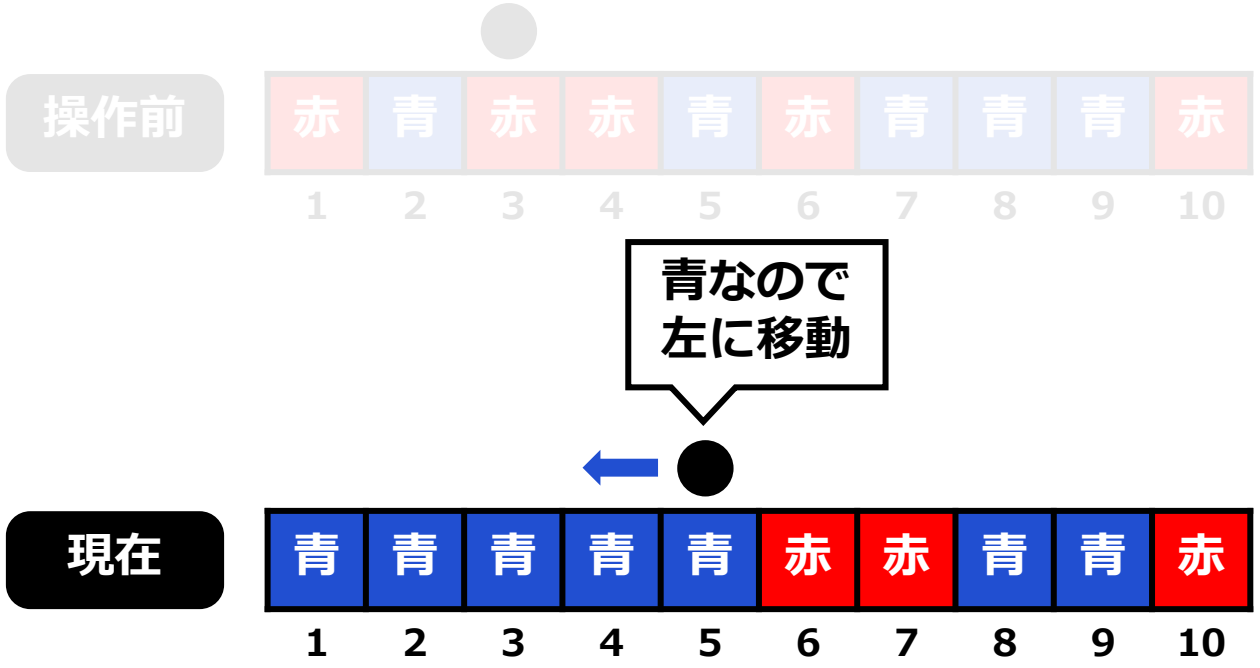
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
青 青 青 青 青 青 青 青 青 赤
1 2 3 4 5 6 7 8 9 10

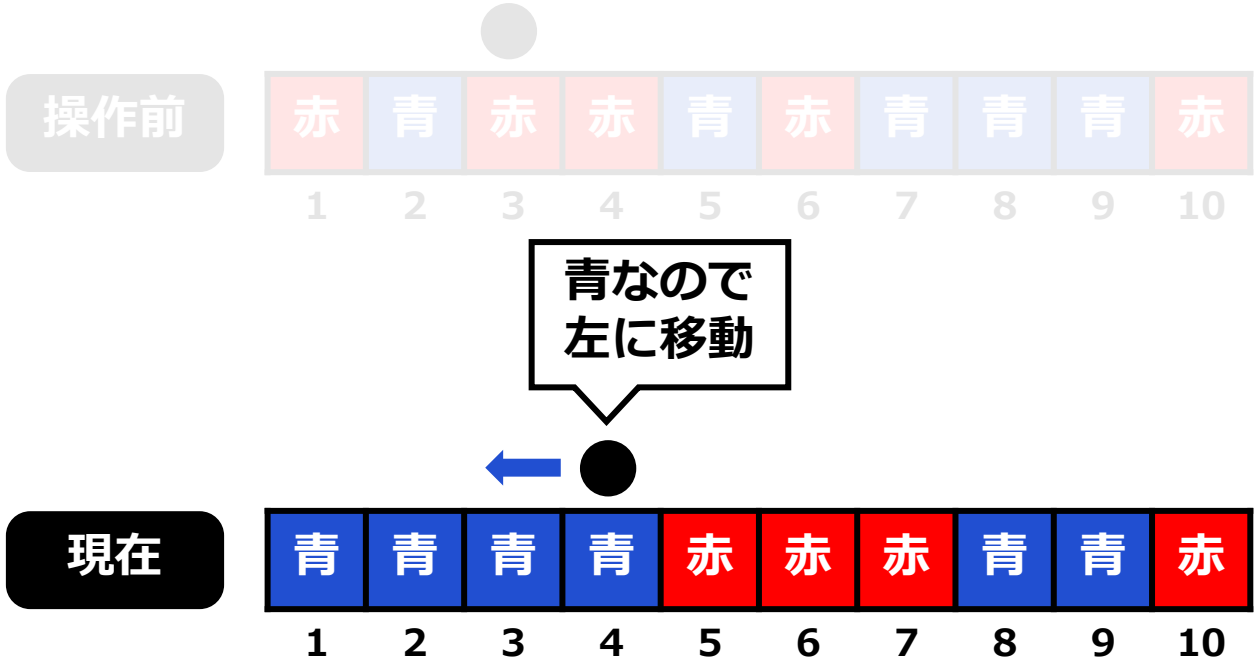
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
青 青 青 青 青 青 赤 青 青 赤
1 2 3 4 5 6 7 8 9 10

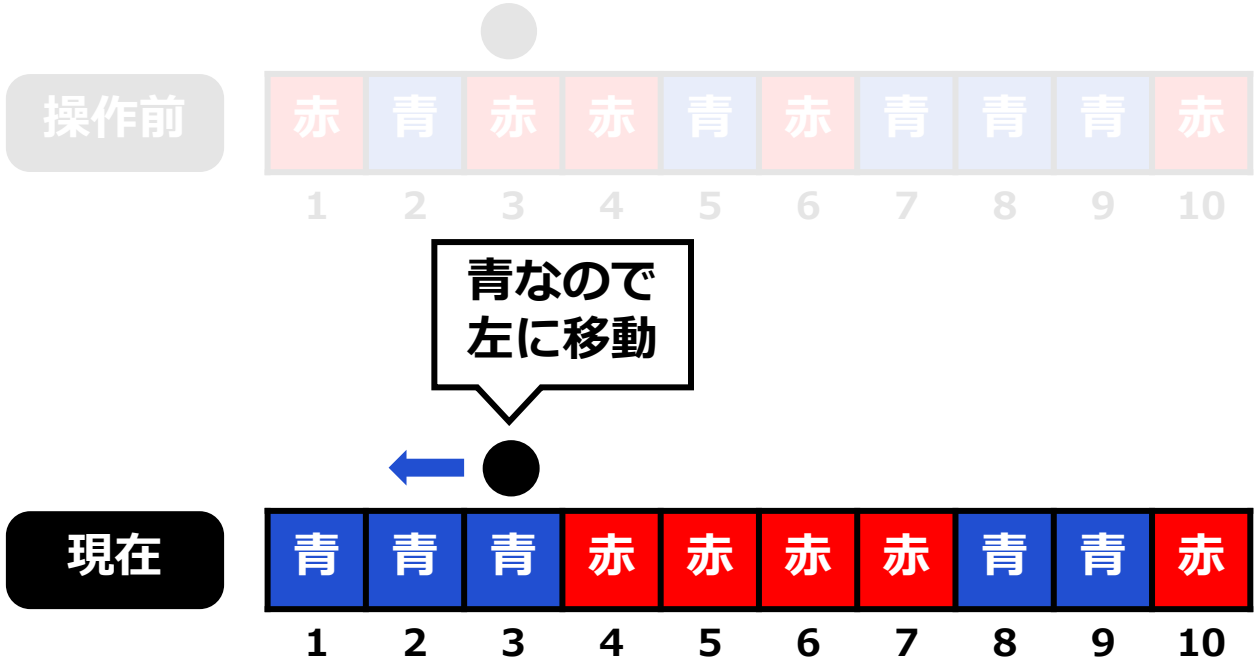
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
青 青 青 青 青 赤 赤 青 青 赤
1 2 3 4 5 6 7 8 9 10

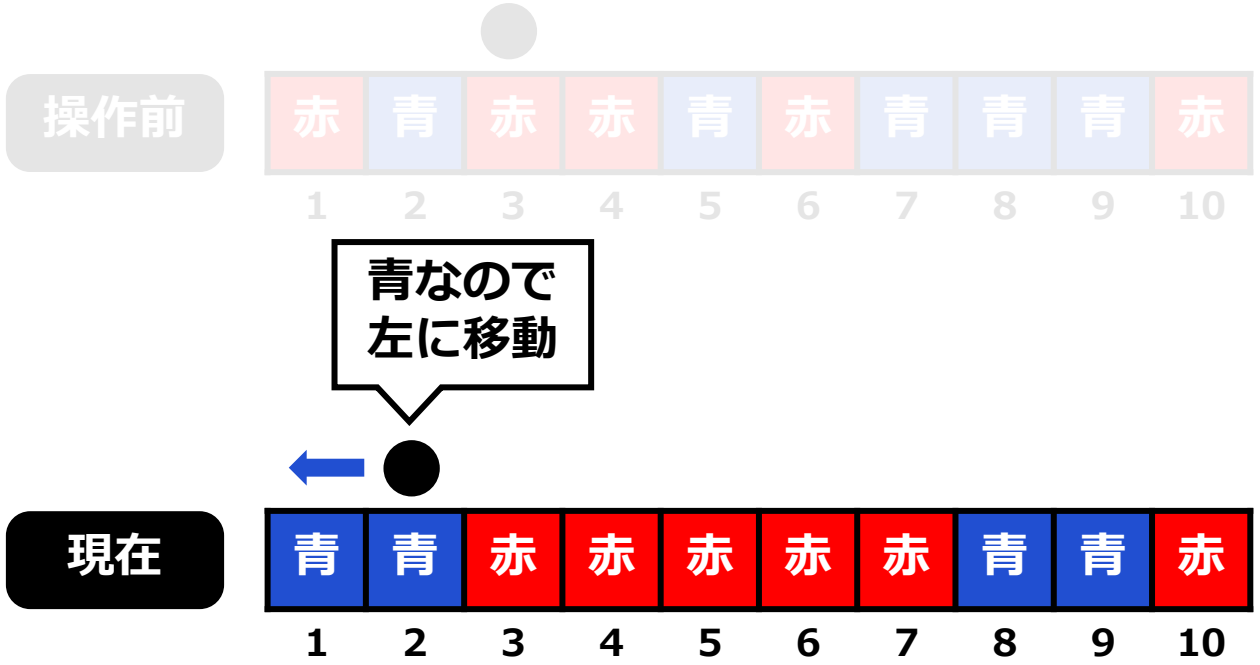
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
青 青 青 青 赤 赤 赤 青 青 赤
1 2 3 4 5 6 7 8 9 10

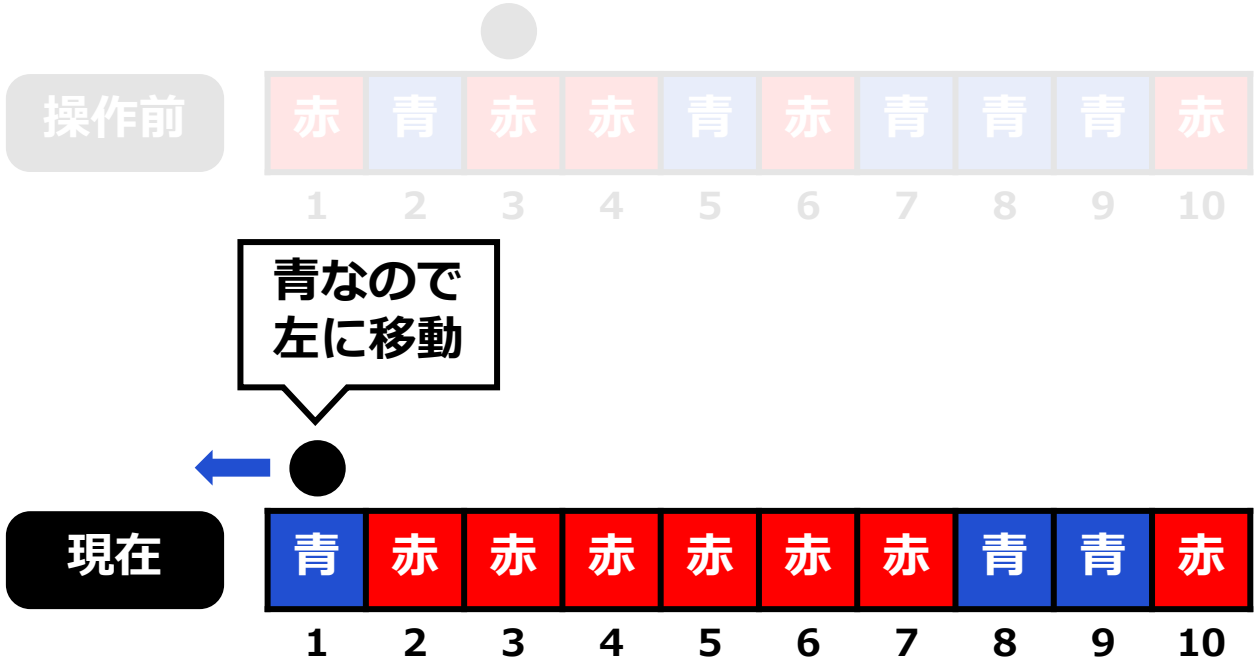
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
青 青 青 赤 赤 赤 赤 青 青 赤
1 2 3 4 5 6 7 8 9 10

2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
青 青 赤 赤 赤 赤 赤 青 青 赤
1 2 3 4 5 6 7 8 9 10

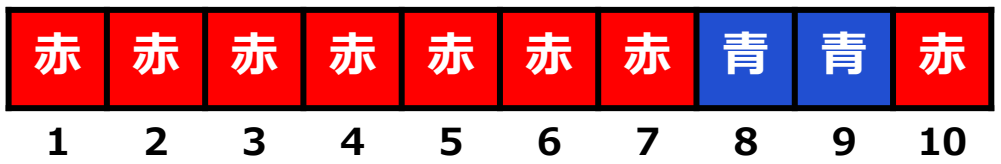
2 重要な考察：具体例


操作前
赤 青 赤 赤 青 赤 青 青 青 赤
1 2 3 4 5 6 7 8 9 10
青なので
左に移動
現在
青 赤 赤 赤 赤 赤 赤 青 青 赤
1 2 3 4 5 6 7 8 9 10

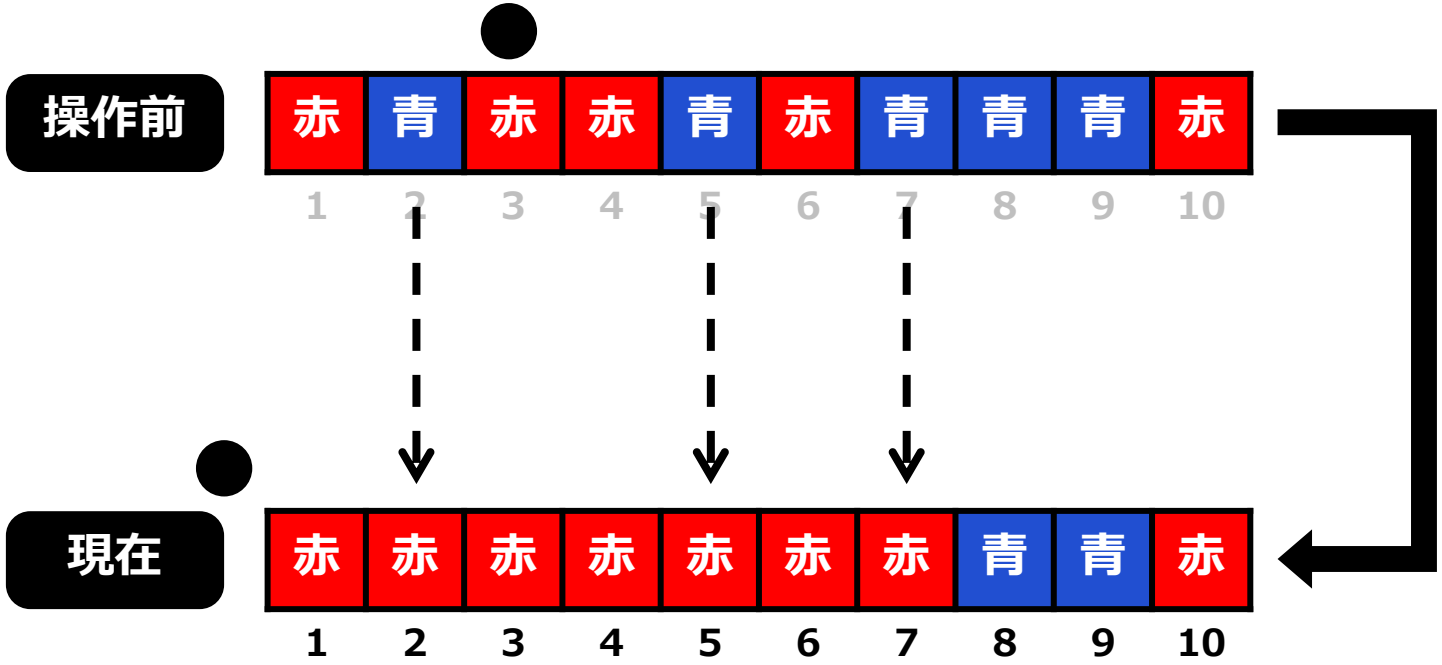
2 重要な考察：具体例

操作前

現在



2 重要な考察：具体例



ボールはタイル 3 から
スタートしたが…

左から 3 つ目までの青タイル
(タイル 2, 5, 7) が赤に！

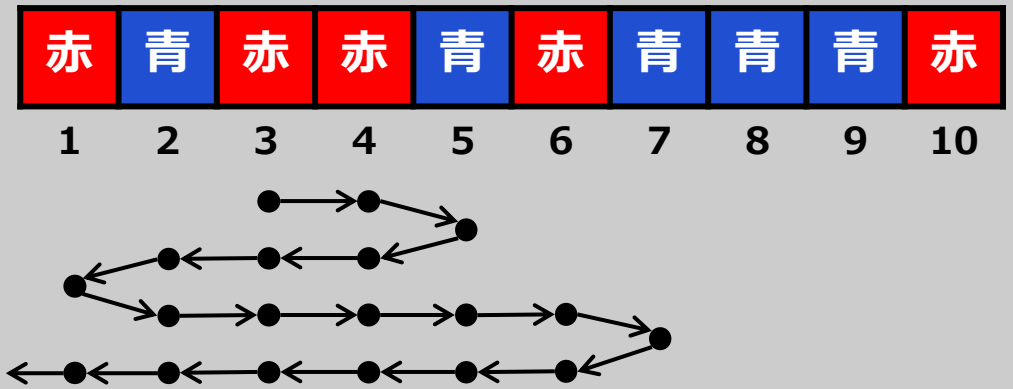
どうしてこんなことになるのか？

まずはボールが左から出る場合を証明しましょう

※証明は少し難しいので、読み飛ばしてもかまいません

2 重要な考察：証明 (1/3)

まず、タイル x にあったボールが左から出た場合、「右に移動する回数」より「左に移動する回数」の方が x 回だけ多い



たとえば先程の例の場合

ボールはタイル 3 から出発するが…

- 右移動：8回
- 左移動：11回 ← 3回多い

2 重要な考察：証明 (2/3)

つまり、「青→赤になったタイル」が「赤→青になったタイル」より

x 個だけ多いことになる

なぜか？

- 青→赤になったタイルについては
  右移動の方が 1 回多い
- 赤→青になったタイルについては
  左移動の方が 1 回多い
- それ以外の場合は左移動 = 右移動

2 重要な考察：証明 (2/3)

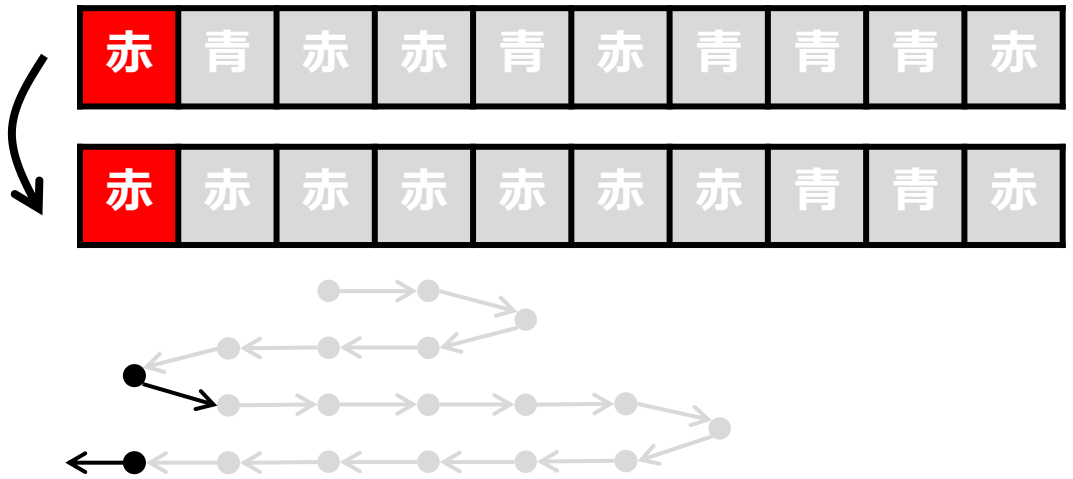
つまり、「青→赤になったタイル」が「赤→青になったタイル」より

x 個だけ多いことになる

なぜか？

- 青→赤になったタイルについては
  右移動の方が 1 回多い
- 赤→青になったタイルについては
  左移動の方が 1 回多い
- それ以外の場合は左移動 = 右移動

タイル1：赤→赤
- 左移動 1 回
- 右移動 1 回



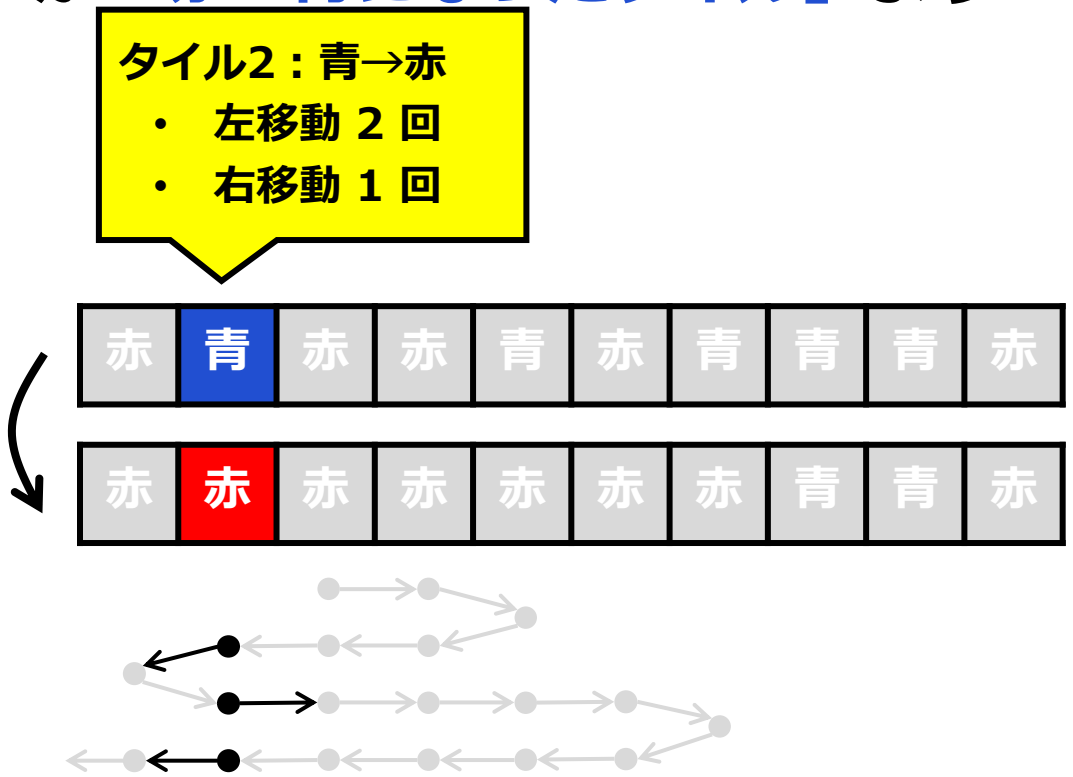
2 重要な考察：証明 (2/3)

つまり、「青→赤になったタイル」が「赤→青になったタイル」より

x 個だけ多いことになる

なぜか？

- 青→赤になったタイルについては
  右移動の方が 1 回多い
- 赤→青になったタイルについては
  左移動の方が 1 回多い
- それ以外の場合は左移動 = 右移動



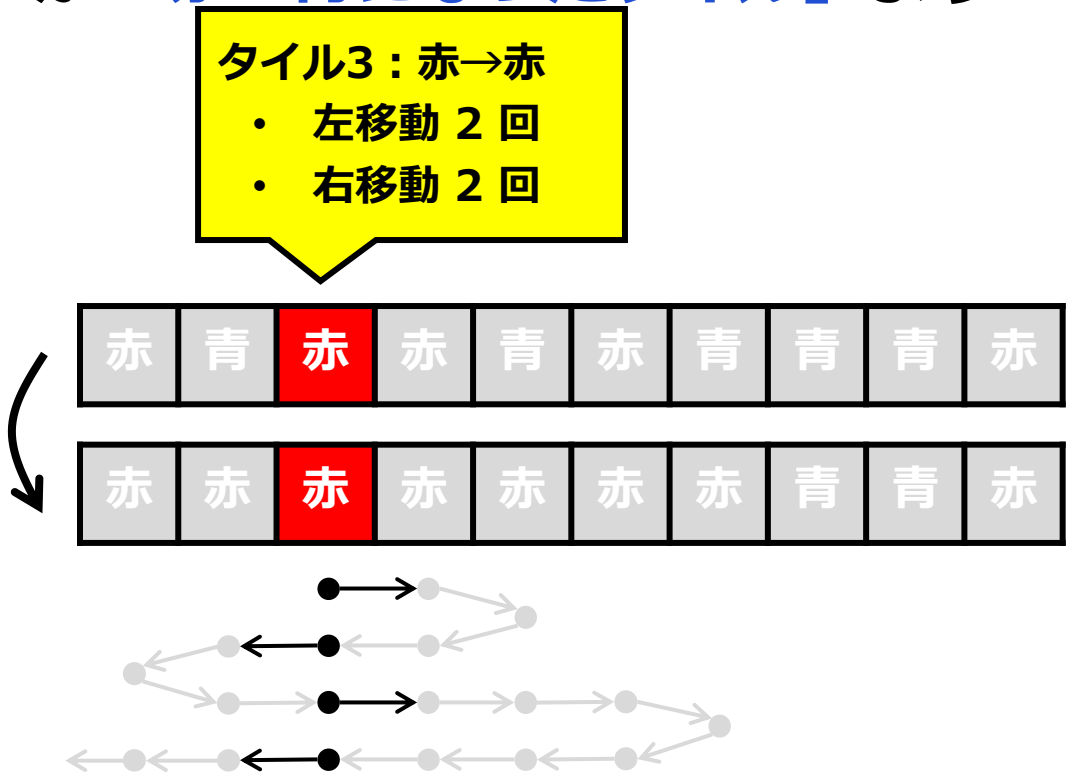
2 重要な考察：証明 (2/3)

つまり、「青→赤になったタイル」が「赤→青になったタイル」より

x 個だけ多いことになる

なぜか？

- 青→赤になったタイルについては
  右移動の方が 1 回多い
- 赤→青になったタイルについては
  左移動の方が 1 回多い
- それ以外の場合は左移動 = 右移動



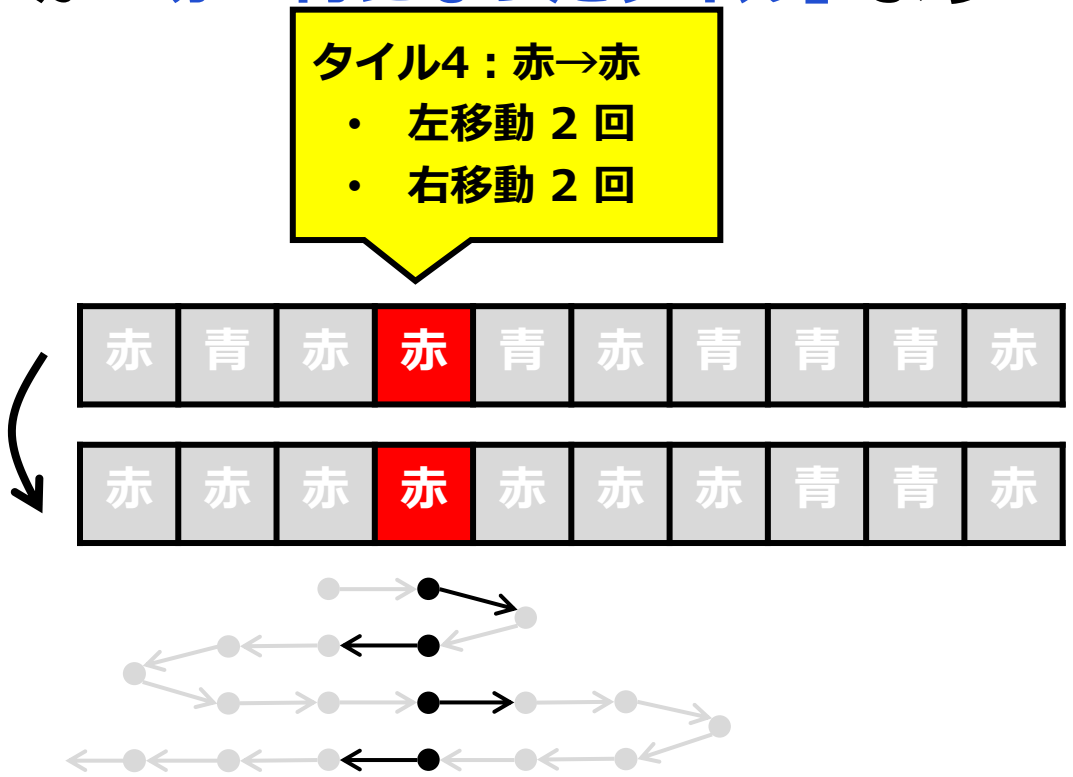
2 重要な考察：証明 (2/3)

つまり、「青→赤になったタイル」が「赤→青になったタイル」より

x 個だけ多いことになる

なぜか？

- 青→赤になったタイルについては
  右移動の方が 1 回多い
- 赤→青になったタイルについては
  左移動の方が 1 回多い
- それ以外の場合は左移動 = 右移動



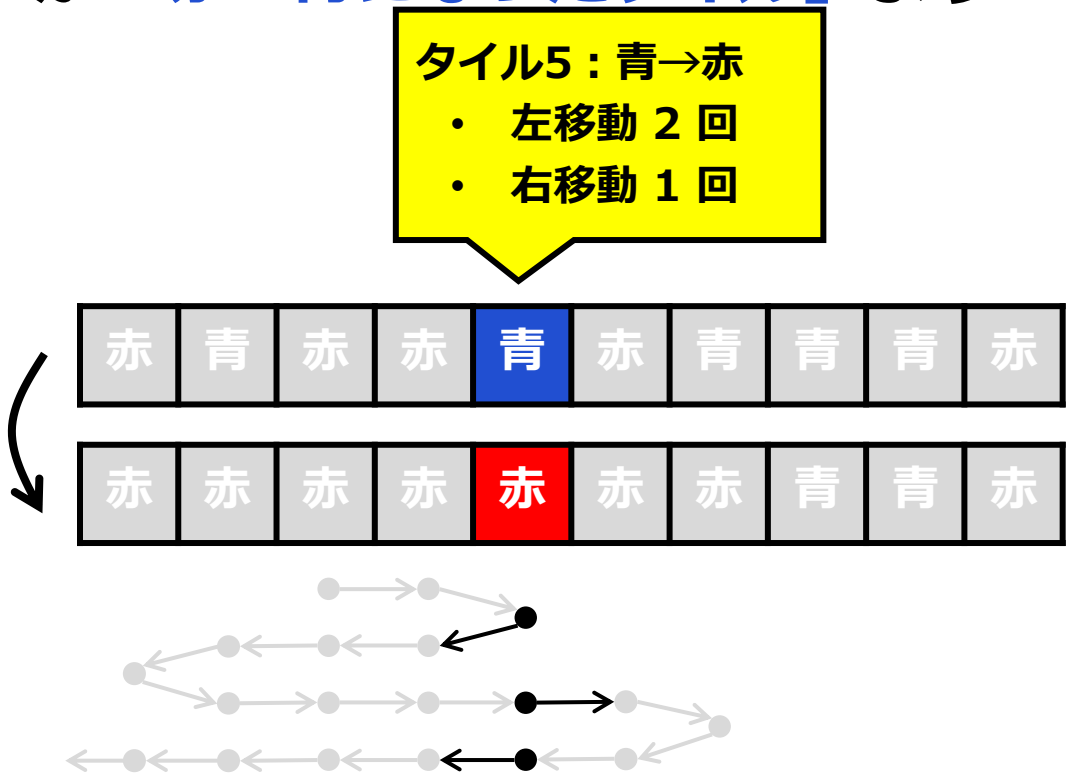
2 重要な考察：証明 (2/3)

つまり、「青→赤になったタイル」が「赤→青になったタイル」より

x 個だけ多いことになる

なぜか？

- 青→赤になったタイルについては
  右移動の方が 1 回多い
- 赤→青になったタイルについては
  左移動の方が 1 回多い
- それ以外の場合は左移動 = 右移動



2 重要な考察：証明 (2/3)

つまり、「青→赤になったタイル」が「赤→青になったタイル」より

x 個だけ多いことになる

なぜか？

- 青→赤になったタイルについては
  右移動の方が 1 回多い
- 赤→青になったタイルについては
  左移動の方が 1 回多い
- それ以外の場合は左移動 = 右移動

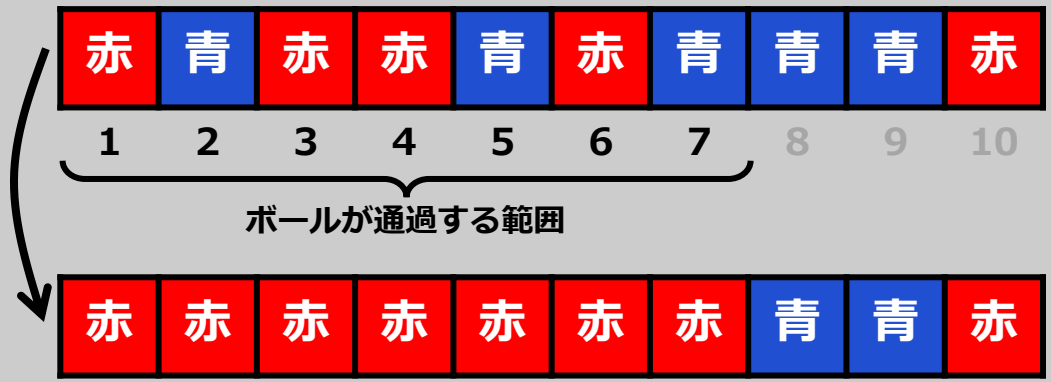
このようになる理由

タイルが青→赤→青→赤→青→赤→青→…
と変化するため

2 重要な考察：証明 (3/3)

そこで、ボールが通った地点は最終的にはすべて赤になる

➡ **左から x 番目までの青タイルが赤に変わるはず！※**



たとえば先程の例の場合

たしかにタイル 7 までが赤になっている

※赤→青になるタイルは存在しないため。そして、左の青タイルから順に青→赤になるため

2 重要な考察：証明 (3/3)

しかし、ボールが通った地点は最終的にはすべて赤になる

➡ 左から x 番目までの青タイルが赤に変わるはず！※

これが、ボールが左から出るとき
左から x 番目までの青だけが赤に変わる理由

たとえば先程の例の場合

※ボールが右から出るときも同じことがいえる

※赤→青になるタイルは存在しないため。そして、左の青タイルから順に青→赤になるため

2 小課題2 (累計15点)

この性質を使って、小課題 2 が解けないか？

2 小課題2（累計15点）

まず、ボールがどちら側から出るかは、以下のように判定できる

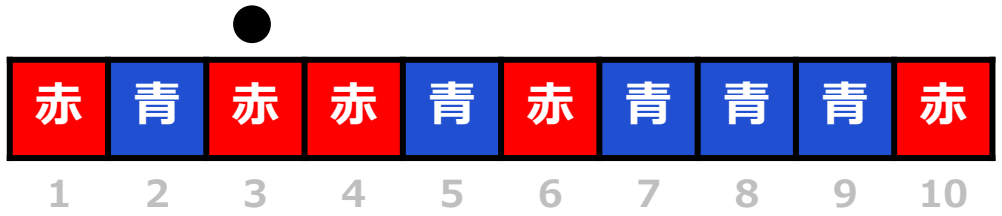
- 青タイルの数が x 個以上の場合 ➡ 左から出る
- 青タイルの数が x 個未満の場合 ➡ 右から出る

※赤タイルの数が x 個未満の場合に限り、青タイルの数が $N+1-x$ 個以上である（= 右から出れる条件を満たす）ことに注意

2 小課題2 (累計15点)

まず、ボールがどちら側から出るかは、以下のように判定できる

- 青タイルの数が x 個以上の場合 ➡ 左から出る
- 青タイルの数が x 個未満の場合 ➡ 右から出る



この例の場合、青タイルが 5 枚あり
3 枚以上なので左から出る

※赤タイルの数が x 個未満の場合に限り、青タイルの数が $N+1-x$ 個以上である（= 右から出れる条件を満たす）ことに注意

2 小課題2 (累計15点)

- これが分かれば、あとは左から x 個 (または右から $N+1-x$ 個) のタイルの色を愚直に変えていけば良い
- ボタン 1 回当たりの計算量は $O(N)$ なので、全体の計算量は $O(NM)$

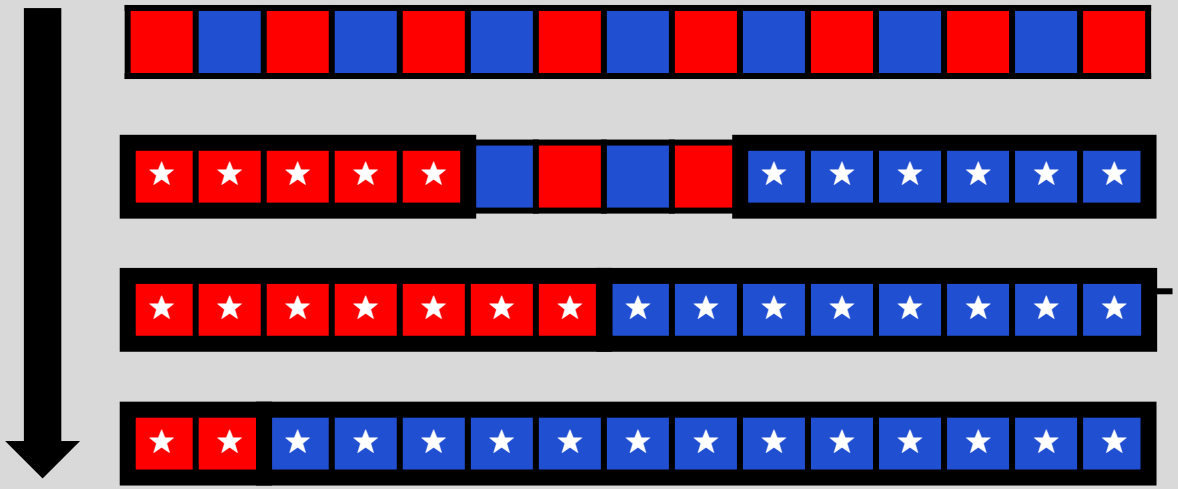
15点

小課題 3

$Q \leq 5$

3 重要な考察

ボタンを何回か押すと、盤面は次のように変化する



フェーズ1

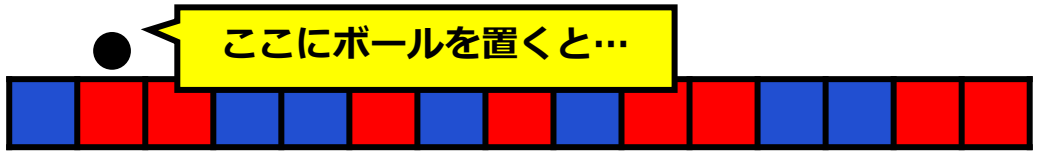
左の赤と右の青の範囲が
徐々に広がっていく

フェーズ2

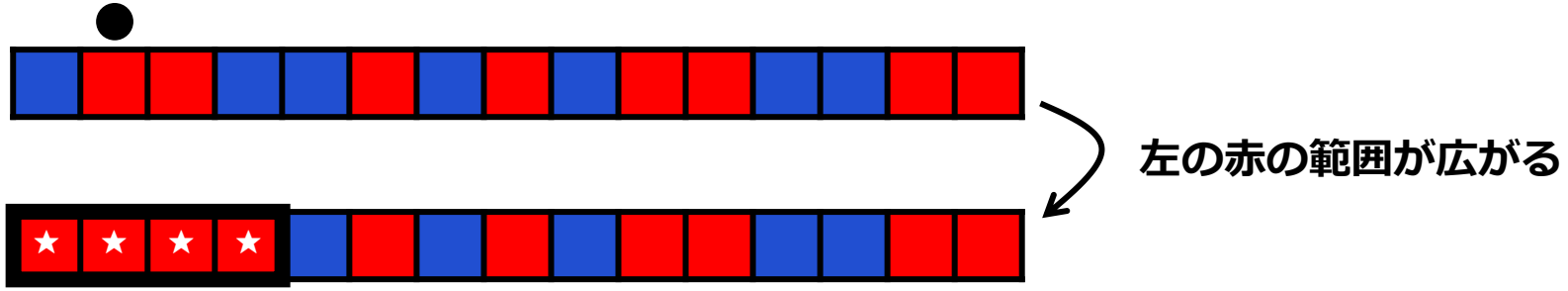
一度「左の赤」と「右の青」がぶつかると
それ以降はぶつかったままである

※このような性質は、小課題 2 の考察からわかる

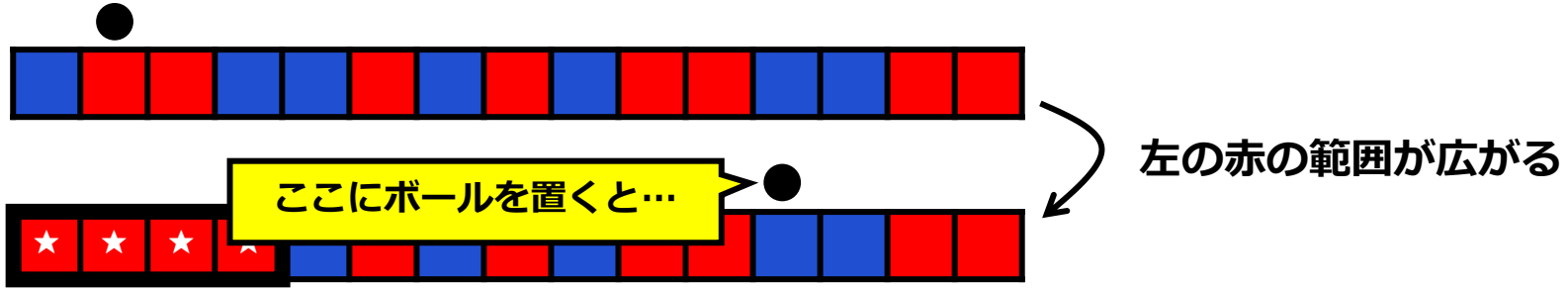
3 重要な考察：例



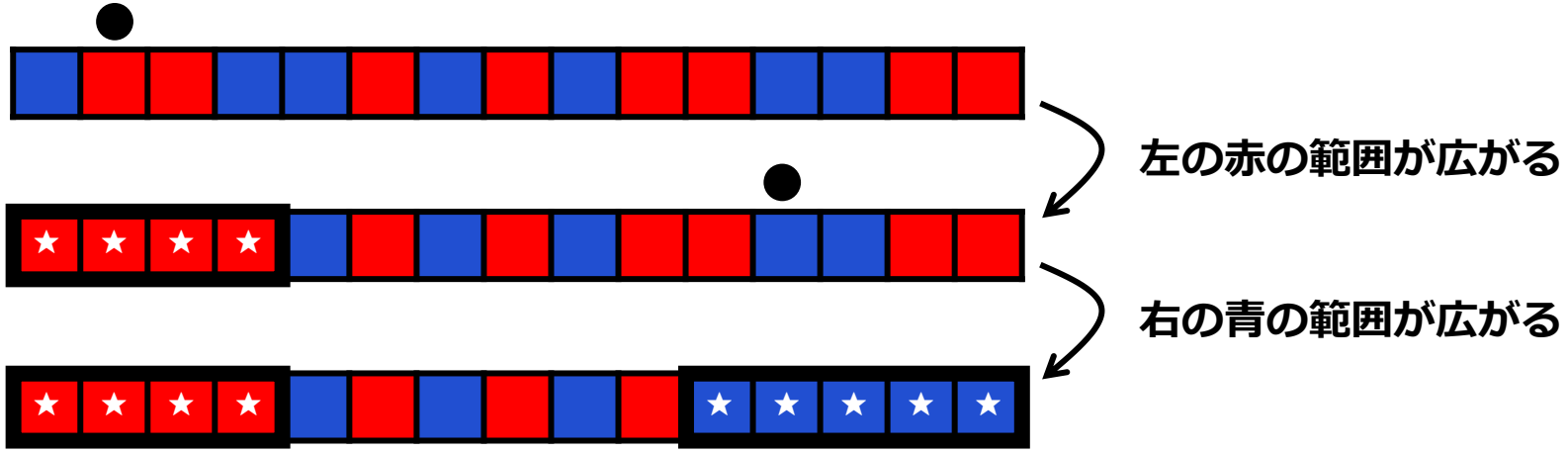
3 重要な考察：例



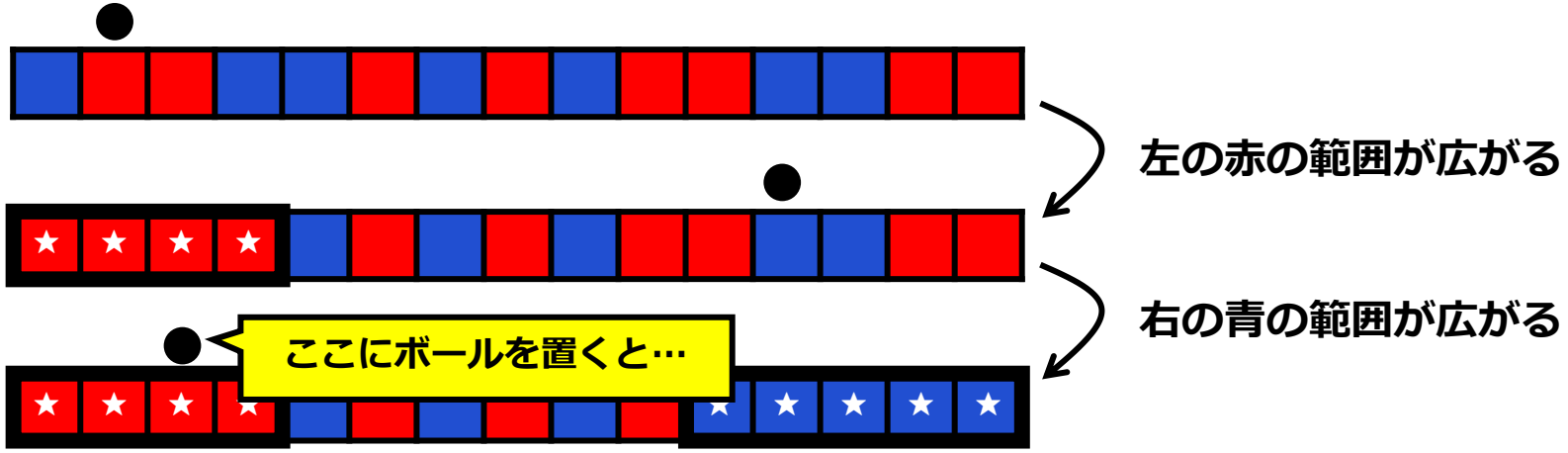
3 重要な考察：例



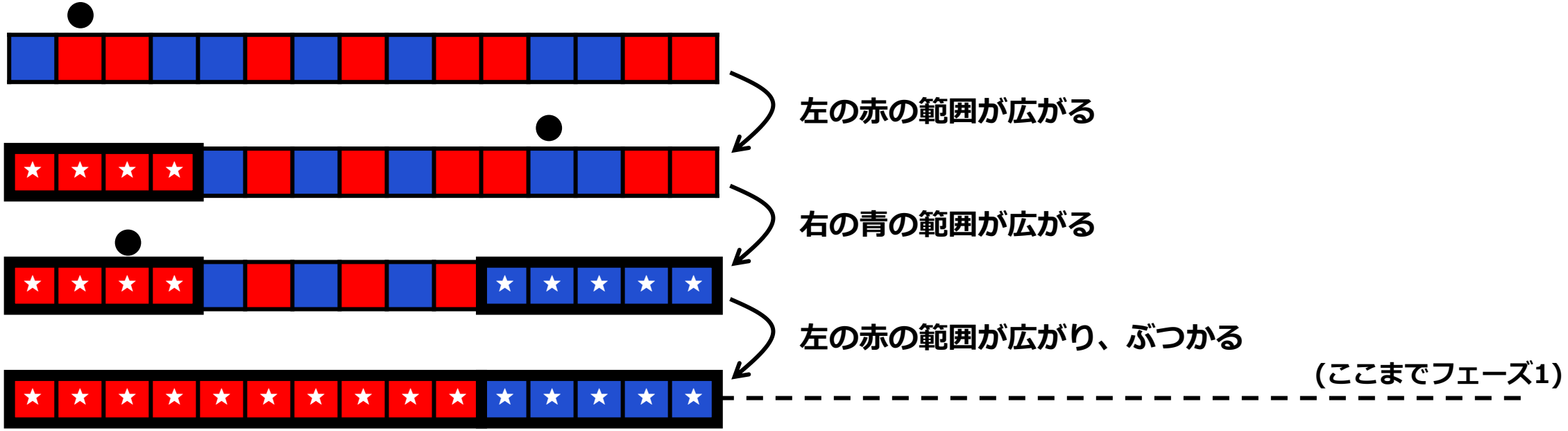
3 重要な考察：例



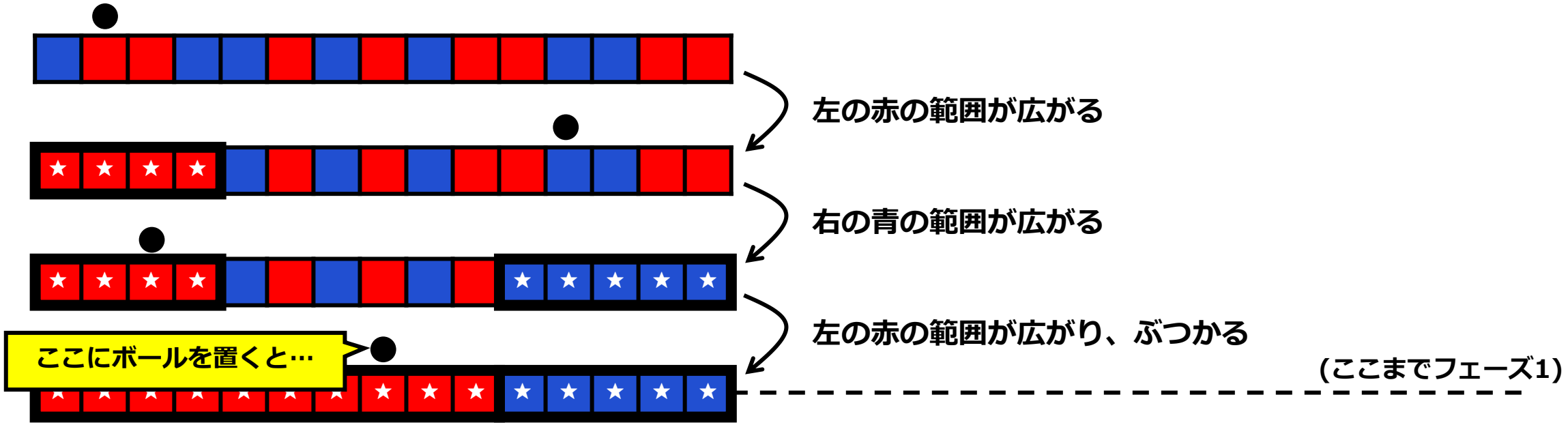
3 重要な考察：例



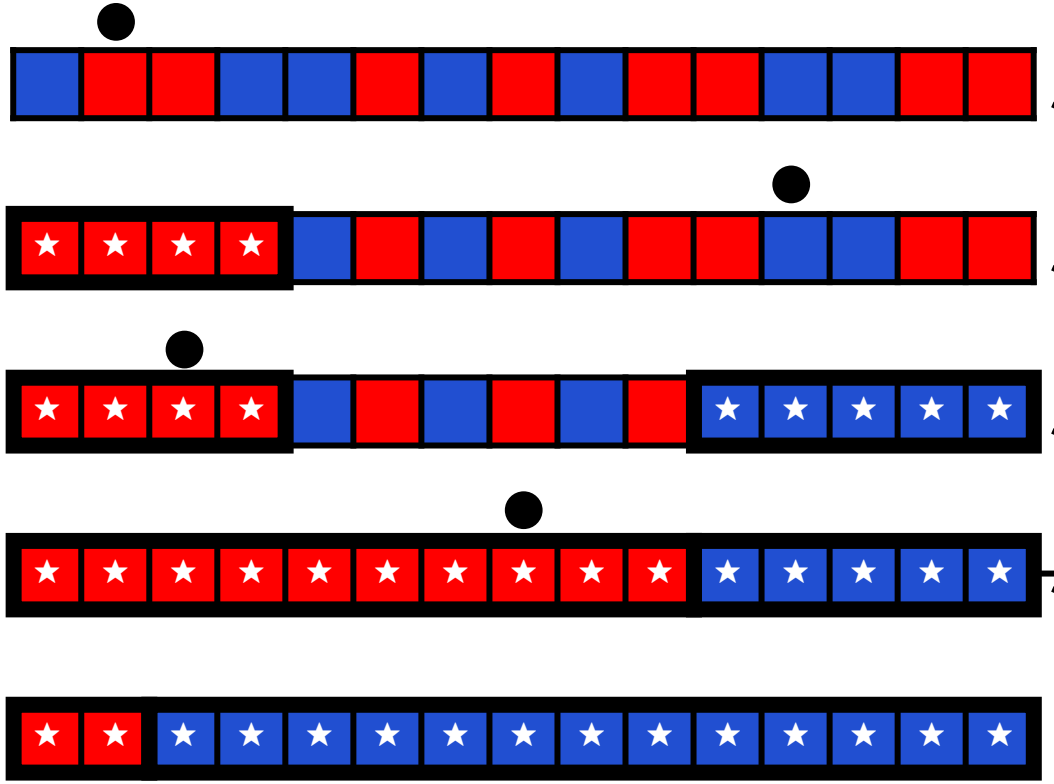
3 重要な考察：例



3 重要な考察：例


左の赤の範囲が広がる
右の青の範囲が広がる
左の赤の範囲が広がり、ぶつかる
ここにボールを置くと…
(ここまでフェーズ1)

3 重要な考察：例



左の赤の範囲が広がる

右の青の範囲が広がる

左の赤の範囲が広がり、ぶつかる

(ここまでフェーズ1)

ぶつかった後は、基本的にぶつかったまま
(赤／青の範囲は増減することがある)

3 小課題3 (累計25点)

この考察を使って

小課題 2 を高速化できないか？

3 小課題3（累計25点）

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

3 小課題3 (累計25点)

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

このフェーズは、以下の 2 つの変数を持つことで

計算量 $O(1)$ で処理できる

- **IndexL：左から何番目までの青タイルが赤になったか**
- **IndexR：右から何番目までの赤タイルが青になったか**

※ぶつかったかどうかの判定は、赤の位置を左から順に保存した配列、青の位置を右から順に保存した配列を用意すると $O(1)$ で行える

3 小課題3 (累計25点)

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

このフェーズは、以下の 2 つの変数を持つことで

計算量 $O(1)$ で処理できる

- IndexL：左から何番目までの青タイルが赤になったか
- IndexR：右から何番目までの赤タイルが青になったか

もしボールをタイル x の上に置いて
左から出た場合
IndexL を x だけ増やせばよい！

※右から出た場合は IndexR を増やす

※ぶつかったかどうかの判定は、赤の位置を左から順に保存した配列、青の位置を右から順に保存した配列を用意すると $O(1)$ で行える

3 小課題3（累計25点）

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

3 小課題3 (累計25点)

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

以下の「すごい性質」を使うと高速化できる

ボールを置いたタイルを赤にした後の赤タイルの枚数を y とすると…

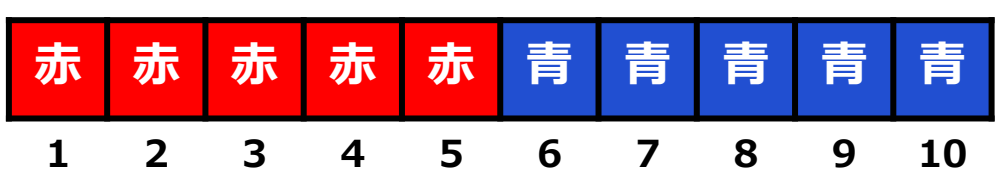
実は、操作後の赤タイルは

$$(x+y) \bmod (N+1) \text{ 枚}$$

※ボールを置く場所をタイル x とする

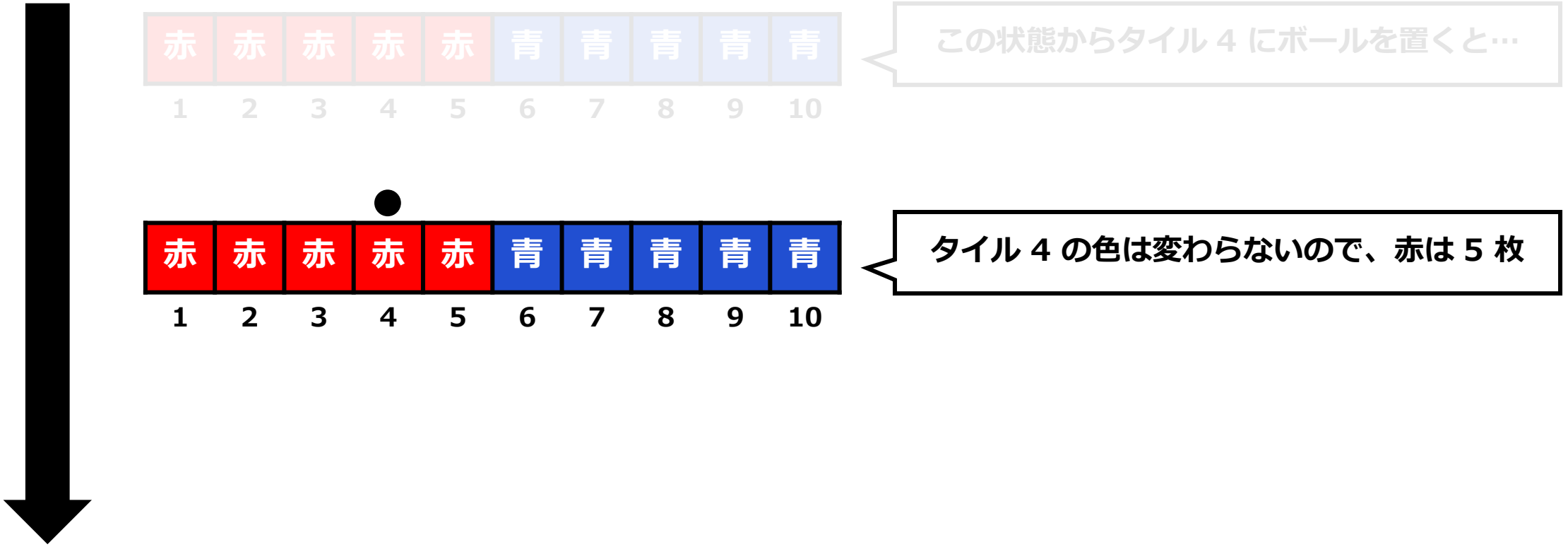
本当か？

3 小課題3のフェーズ2：例A



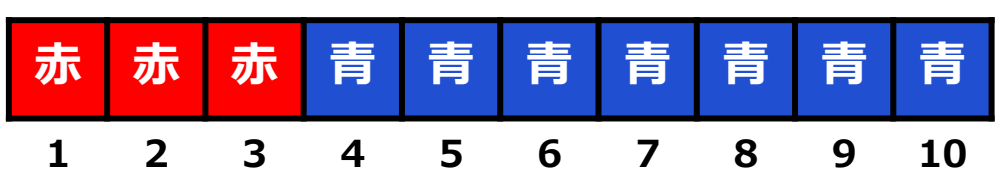
この状態からタイル 4 にボールを置くと…

3 小課題3のフェーズ2：例A



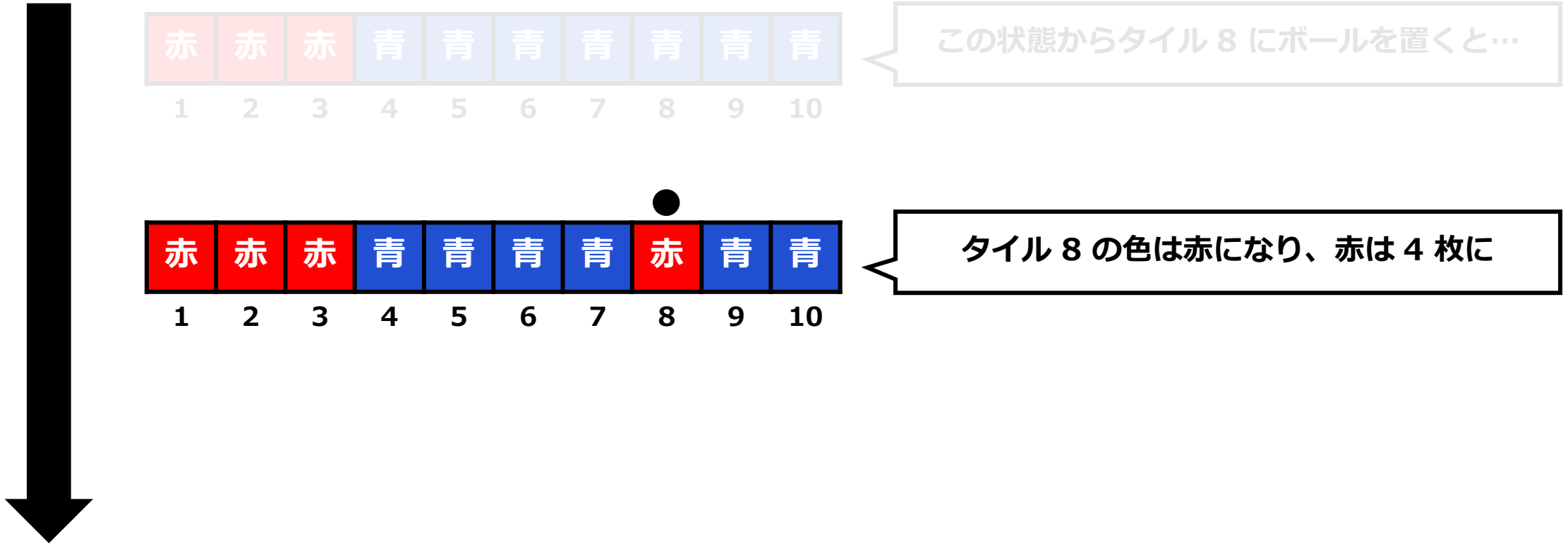
3 小課題3のフェーズ2：例A

3 小課題3のフェーズ2：例B



この状態からタイル 8 にボールを置くと…

3 小課題3のフェーズ2：例B



3 小課題3のフェーズ2：例B

3 小課題3 (累計25点)

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

したがって、ボールの置き場所を x、ボタンを押す前の赤のタイル数を y とするとき
操作後の赤のタイル数は次のようになる

- **$x < y$ のとき：$(x + y + 1) \bmod (N + 1)$**
- **$x \geq y$ のとき：$(x + y) \bmod (N + 1)$**

3 小課題3 (累計25点)

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

したがって、ボールの置き場所を x、ボタンを押す前の赤のタイル数を y とするとき
操作後の赤のタイル数は次のようになる

- **$x < y$ のとき：$(x + y + 1) \bmod (N + 1)$**
- **$x \geq y$ のとき：$(x + y) \bmod (N + 1)$**

+1 が付く理由：
ボールが青タイルの上に置かれるので、
ボールが動く前に赤が 1 つ増えるため

3 小課題3 (累計25点)

この解法では、ボタン 1 回当たり計算量 $O(1)$ で済む

- ボタン M 個、クエリ Q 個なので全体の計算量は $O(MQ)$

25点

3 前半のまとめ

小課題1

愚直にシミュレーションして解く

小課題2

次の性質を使って解く：

- ボールが左から出る場合、左から x 番目までの青タイルが赤になる
- ボールが右から出る場合、右から $N+1-x$ 番目までの赤タイルが青になる

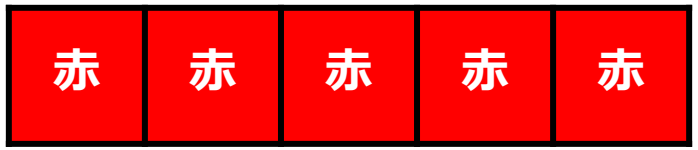
小課題3

- 前半パートは、左から何番目までの青が赤になったか IndexL、右から何番目までの赤が青になったか IndexR を記録して解く
- 後半パートは、赤タイルの枚数が $\mathrm{mod}\ (N+1)$ で簡単に計算できる性質を使う

小課題 4

$N = 10, C_i =$ ‘R’ である

4 小課題4 (累計36点)



最初は全部赤タイルなので、フェーズ 1 はない

➡ **いきなり小課題 3 のフェーズ 2 の解法が使える**

4 小課題4 (累計36点)

したがって、以下のような単純な解法で解くことができる

```
Answer = 5
for (int i = L; i <= R; i++) {
    if (Answer < A[i]) {
        Answer = (Answer + A[i] + 1) % (N + 1);
    }
    else {
        Answer = (Answer + A[i]) % (N + 1);
    }
}
```

4 小課題4 (累計36点)

したがって、以下のような単純な解法で解くことができる

```
Answer = 5
for (int i = L; i <= R; i++) {
    if (Answer < A[i]) {
        Answer = (Answer + A[i] + 1) % (N + 1);
    }
    else {
        Answer = (Answer + A[i]) % (N + 1);
    }
}
```

しかし、$M \leq 120000, Q \leq 120000$

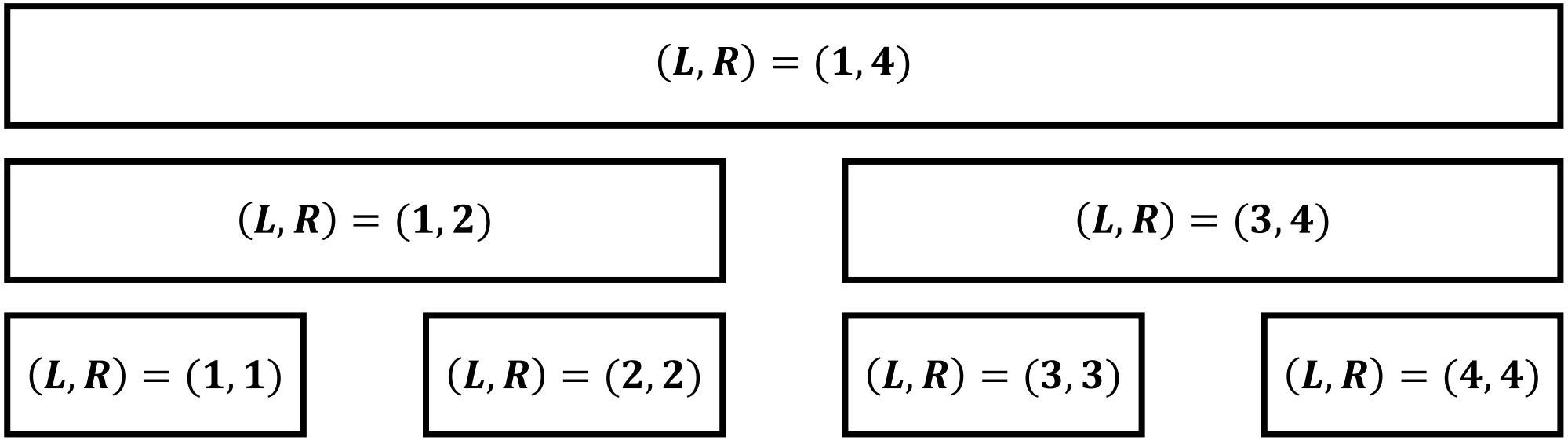
なので TLE してしまう！

4 小課題4 (累計36点)

100 / 191

セグメント木を使って高速化しよう！

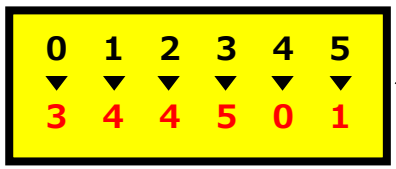
4 小課題4 (累計36点)



各ノードには、ボタン $L, L+1, \ldots, R$ を連続して押したときの

「操作前と操作後のタイルの枚数の関係」を持つ

4 小課題4（累計36点）



ボタン 1 ではボールの位置が 2 なので…

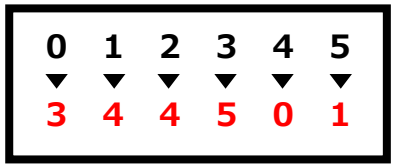
- $x < 2$ の場合：$(x + 3) \bmod 6$
- $x \geq 2$ の場合：$(x + 2) \bmod 6$

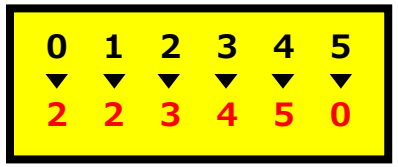
例：操作前が 0 枚の場合、操作後は 3 枚

たとえば、$N = 5, A = [2, 1, 5, 4]$ の場合は上のようになる

※制約は $N = 10$ だが、説明の都合上 $N = 5$ の例を説明している

4 小課題4 (累計36点)





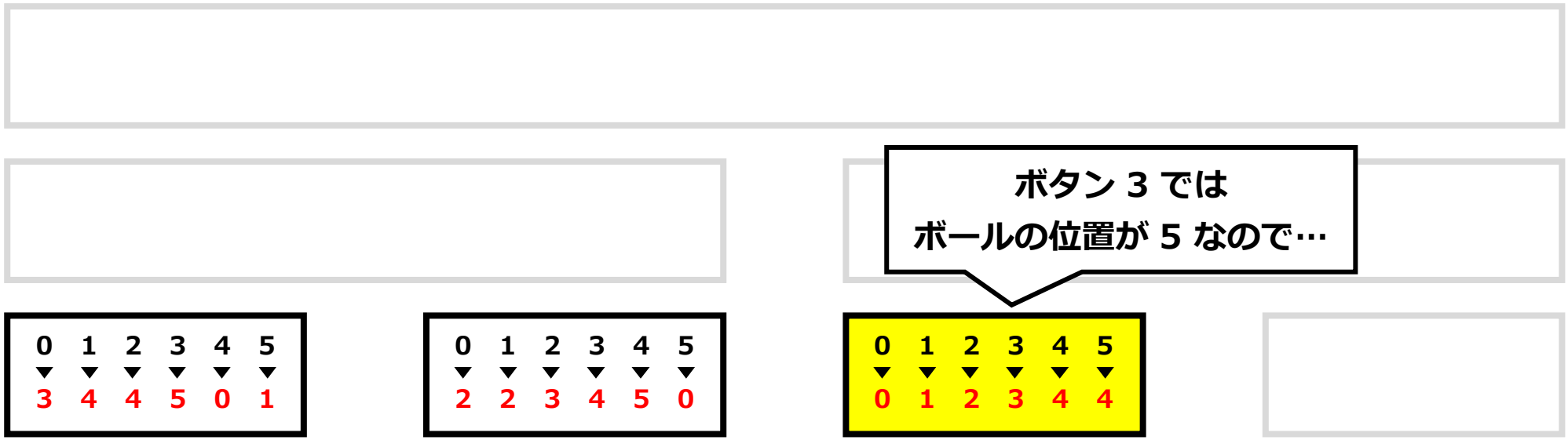
ボタン 2 ではボールの位置が 1 なので…

- $x < 1$ の場合：$(x + 2) \bmod 6$
- $x \geq 1$ の場合：$(x + 1) \bmod 6$

例：操作前が 0 枚の場合、操作後は 2 枚

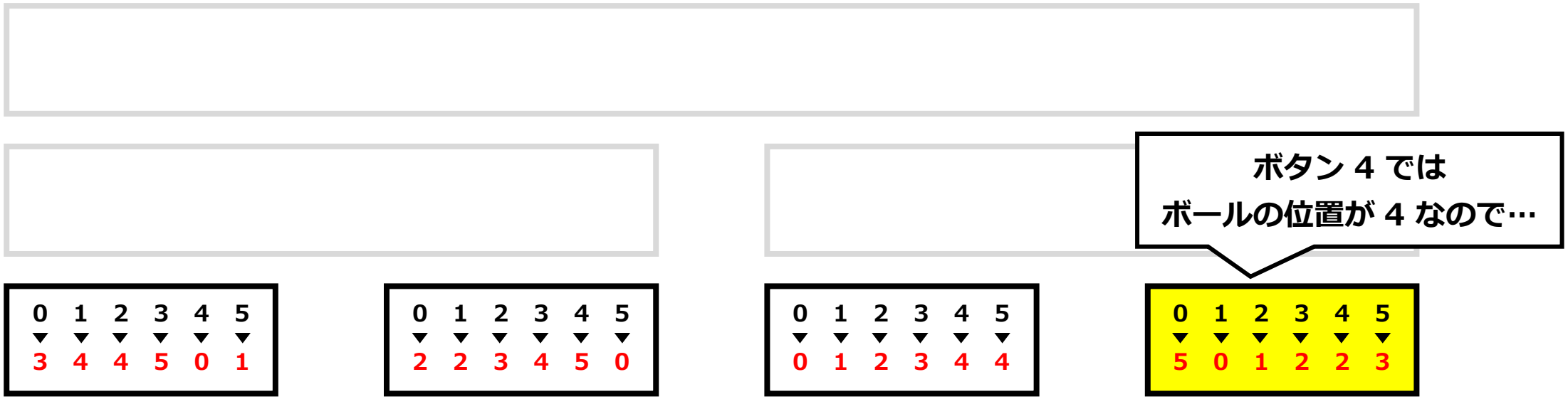
たとえば、$N = 5, A = [2, 1, 5, 4]$ の場合は上のようになる

4 小課題4 (累計36点)



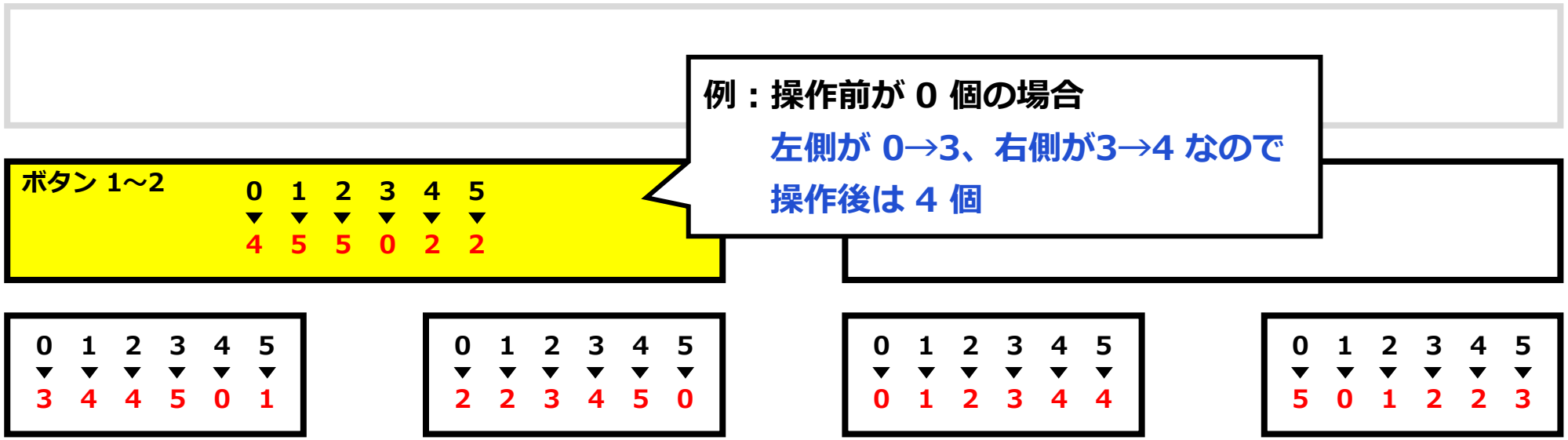
たとえば、$N = 5, A = [2, 1, 5, 4]$ の場合は上のようになる

4 小課題4（累計36点）



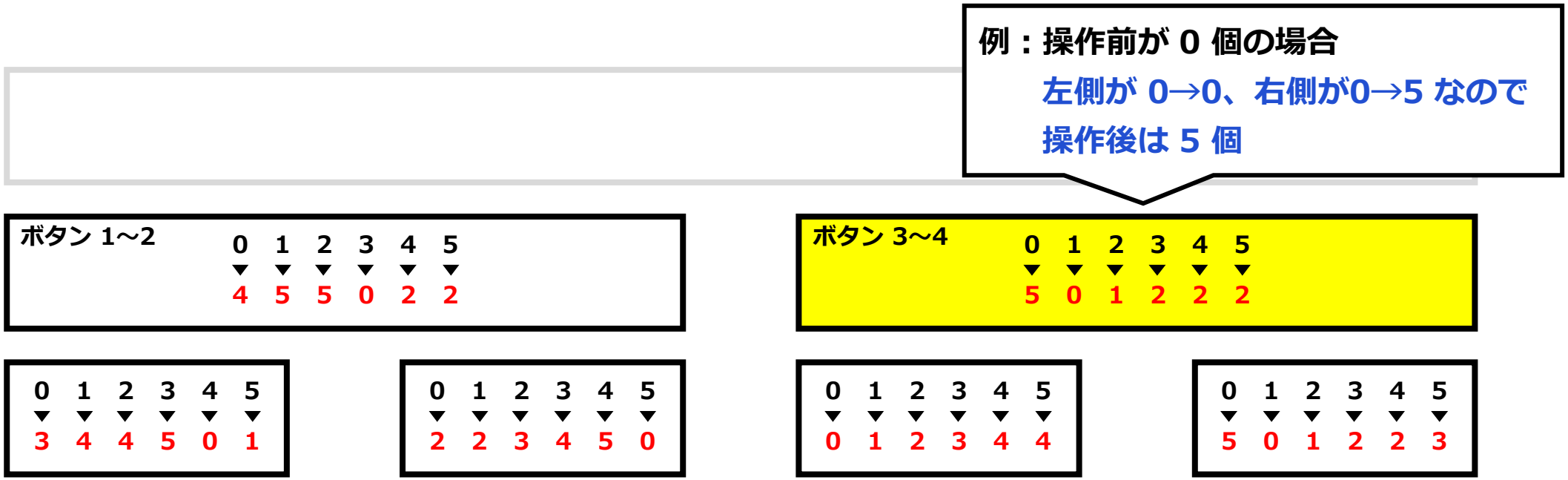
たとえば、$N = 5, A = [2, 1, 5, 4]$ の場合は上のようになる

4 小課題4 (累計36点)



たとえば、$N = 5, A = [2, 1, 5, 4]$ の場合は上のようになる

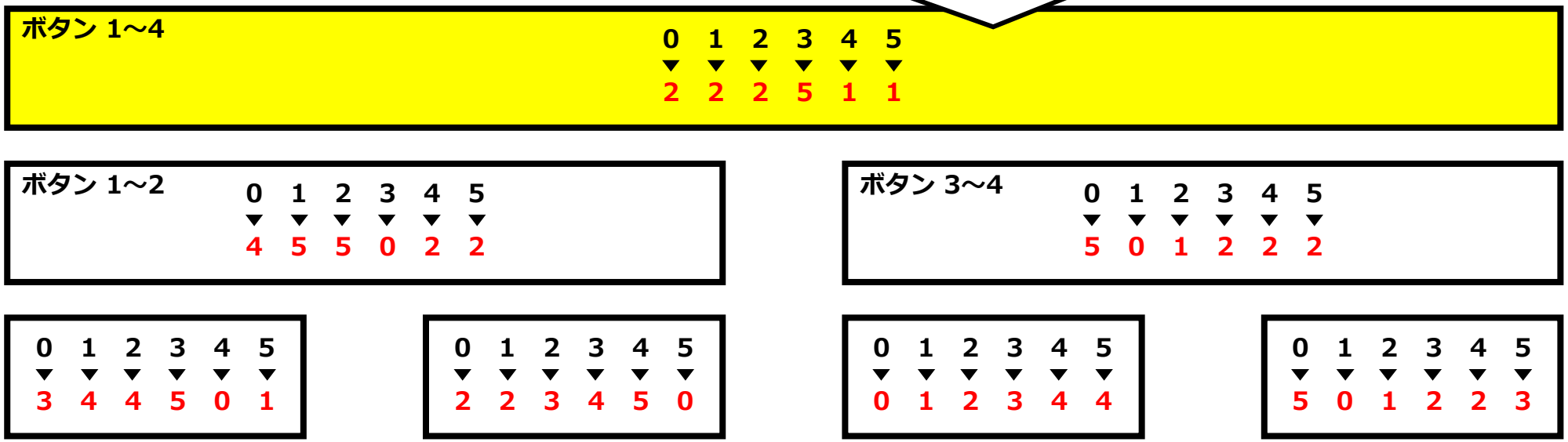
4 小課題4（累計36点）



たとえば、$N = 5, A = [2, 1, 5, 4]$ の場合は上のようになる

4 小課題4（累計36点）

例：操作前が 0 個の場合
左側が 0→4、右側が4→2 なので
操作後は 2 個



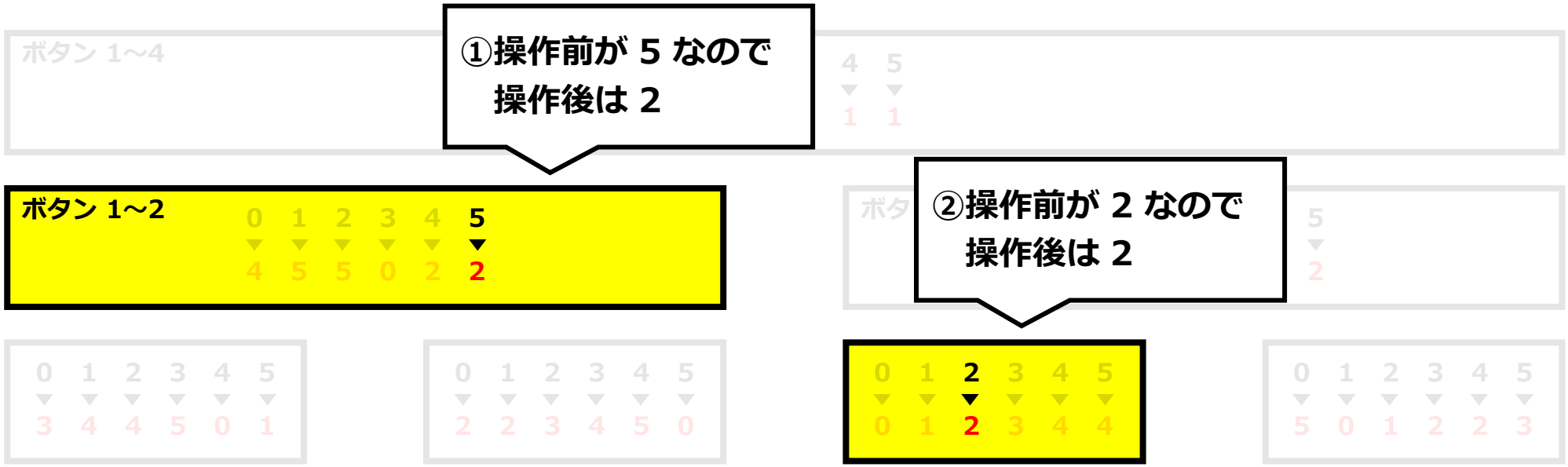
たとえば、$N = 5, A = [2, 1, 5, 4]$ の場合は上のようになる

セグメント木の構成が終わった

次はどうやってクエリに答えるか？

いくつかの例を考えよう

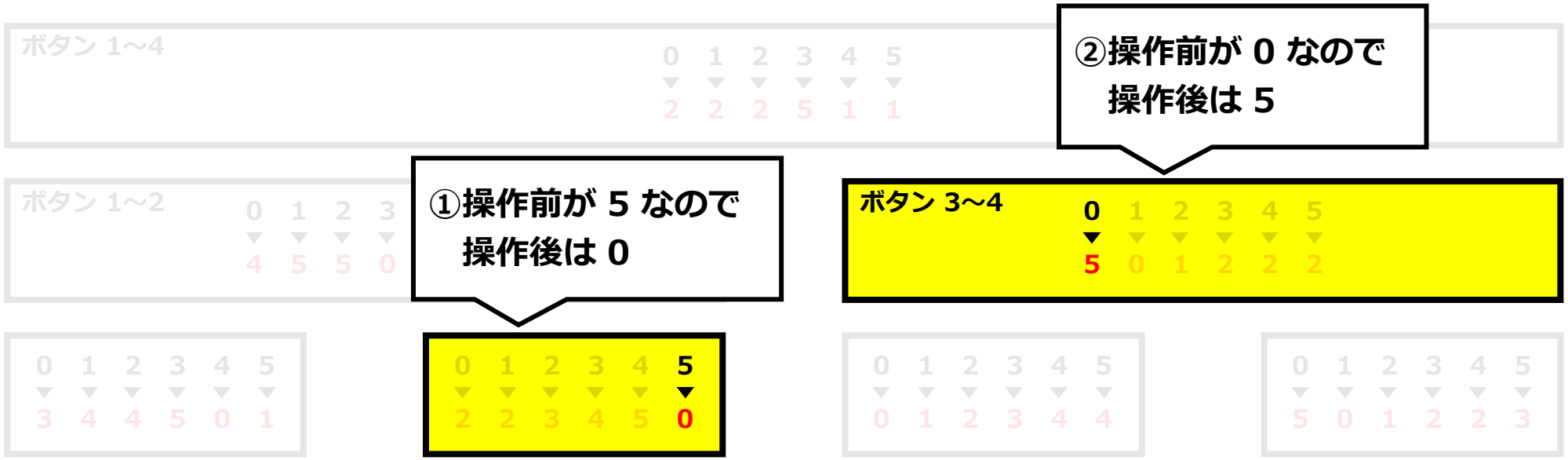
4 小課題4 (累計36点)



たとえば、$(L, R) = (1, 3)$ の場合は $[1, 2]$ と $[3, 3]$ の和で表せるので

最初 5 個の場合、上図のように答えが「2 個」とわかる

4 小課題4 (累計36点)



たとえば、$(L, R) = (2, 4)$ の場合は $[2, 2]$ と $[3, 4]$ の和で表せるので

最初 5 個の場合、上図のように答えが「5 個」とわかる

4 小課題4 (累計36点)

一般の場合でも、任意の区間 $[L, R]$ はセグメント木の $O(\log M)$ 個のノードで表せる

➡ クエリ当たり $O(\log M)$ で答えられる！

セグメント木の構成にかかる時間は $O(M)$ なので、全体の計算量は $O(M + Q \log M)$

36点

小課題 5

C_i が RR…RRBB…BB の形になっている

5 小課題5の問題点

前の小課題は $N = 10$ だったので

セグメント木を直接管理できた

but $N \leq 120000$ だと MLE する！

5 小課題5の問題点

前の小課題は $N = 10$ だったので

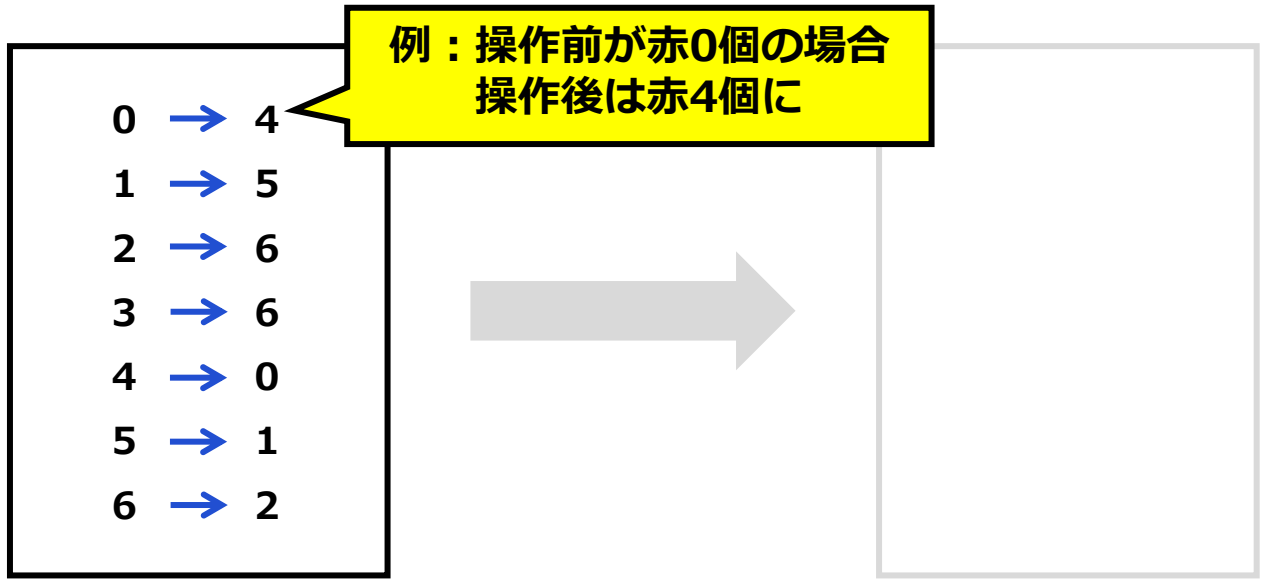
セグメント木を使えました

しかし、「増分が同じになる区間」を圧縮すれば

MLE を回避することができる！

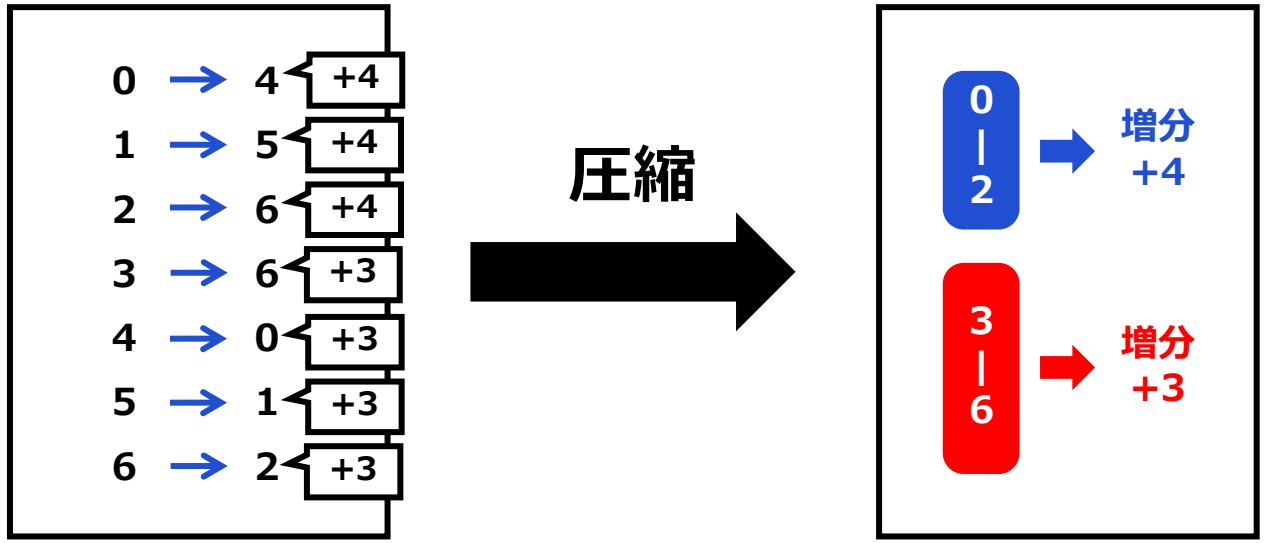
but $N \leq 120000$ だと MLE する！

5 セグ木を圧縮：例1



たとえば、ボールを位置 3 に押した場合の状態遷移は左図

5 セグ木を圧縮：例1

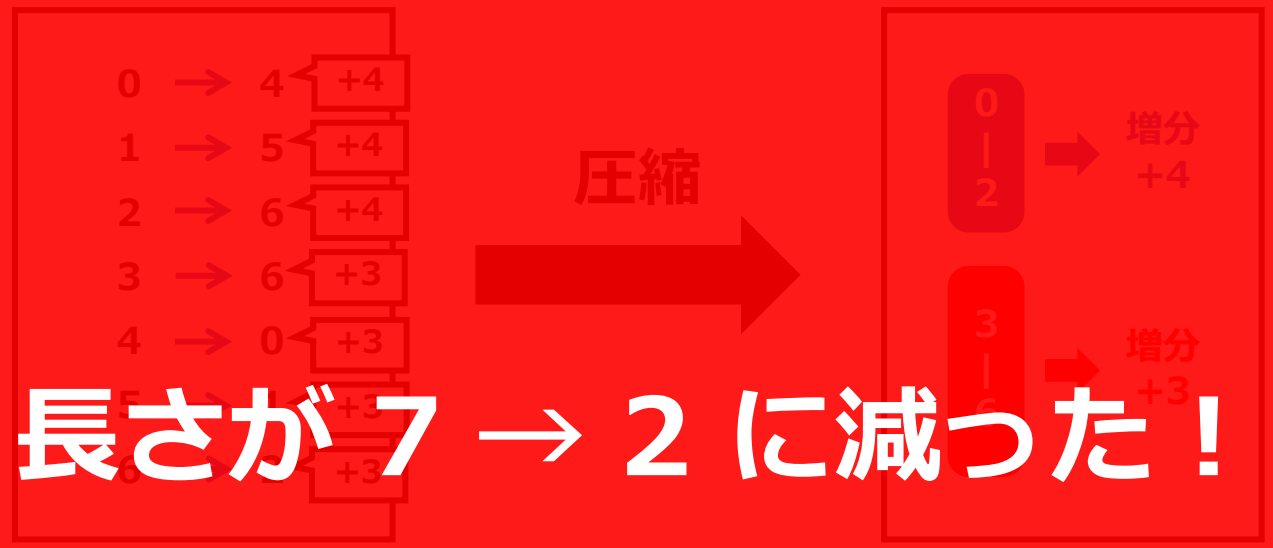


たとえば、ボールを位置 3 に押した場合の状態遷移は左図

これを圧縮すると右図のようになる

※「+4」「+3」などの増分は mod $N+1$ 上での話

5 セグ木を圧縮：例1

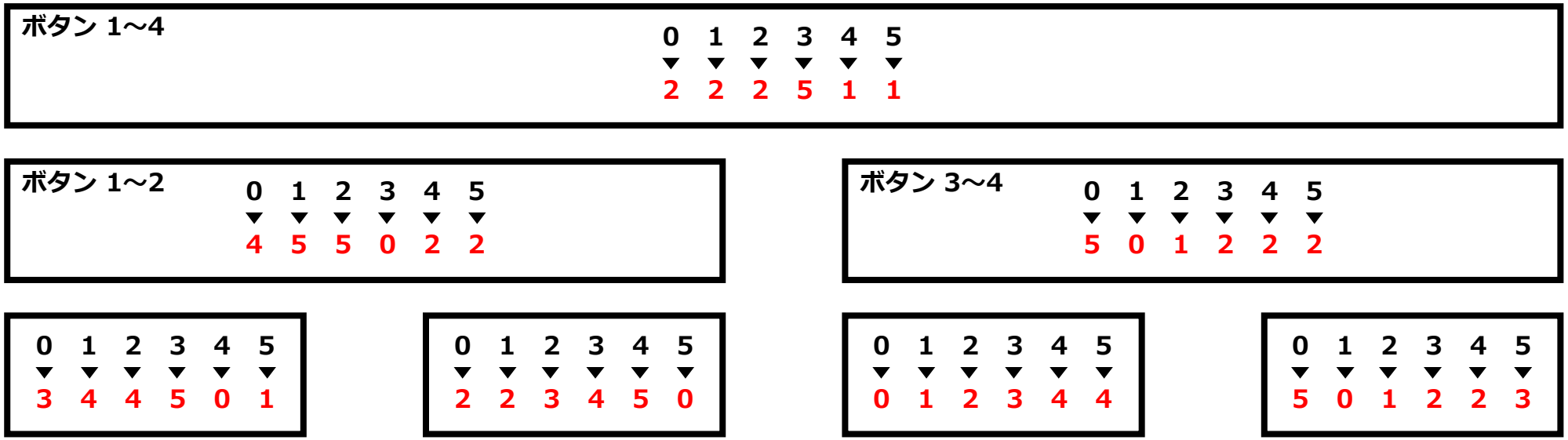


長さが 7 → 2 に減った！

たとえば、ボールを位置 3 に押した場合の状態遷移は左図

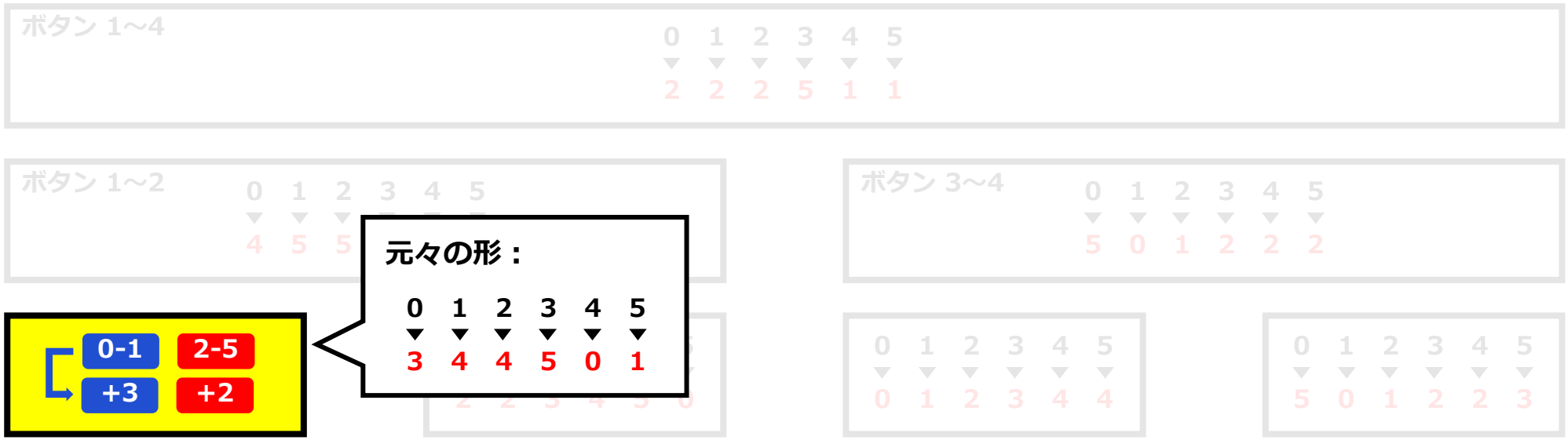
これを圧縮すると右図のようになる

5 セグ木を圧縮：例2



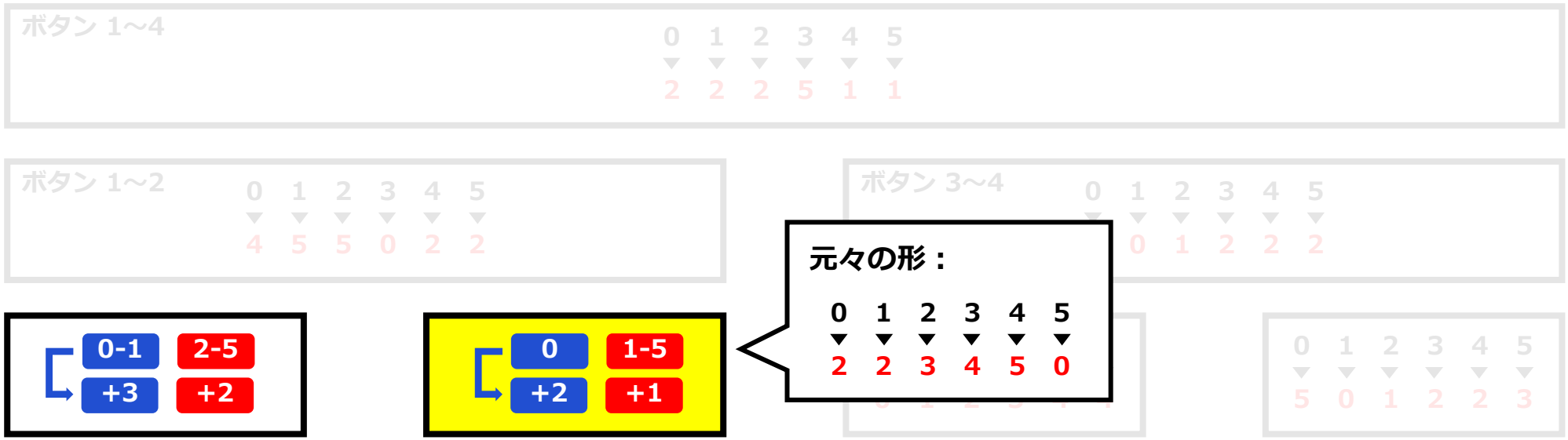
たとえば、$N = 4, A = [2, 1, 5, 4]$ のときのセグメント木は上図の通り

5 セグ木を圧縮：例2



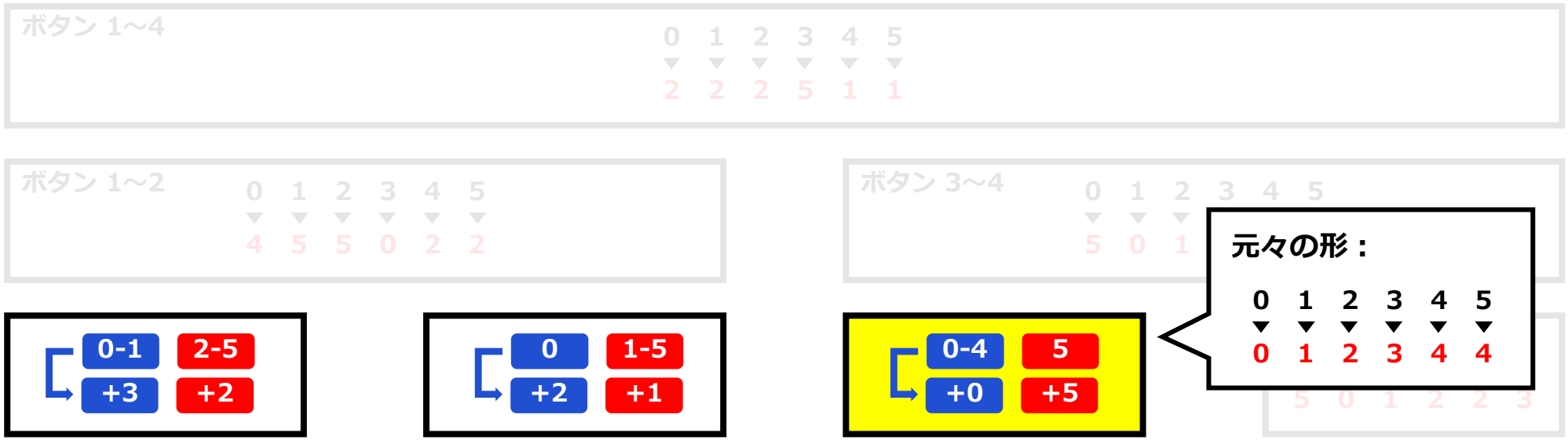
これを圧縮すると、このように変化する

5 セグ木を圧縮：例2



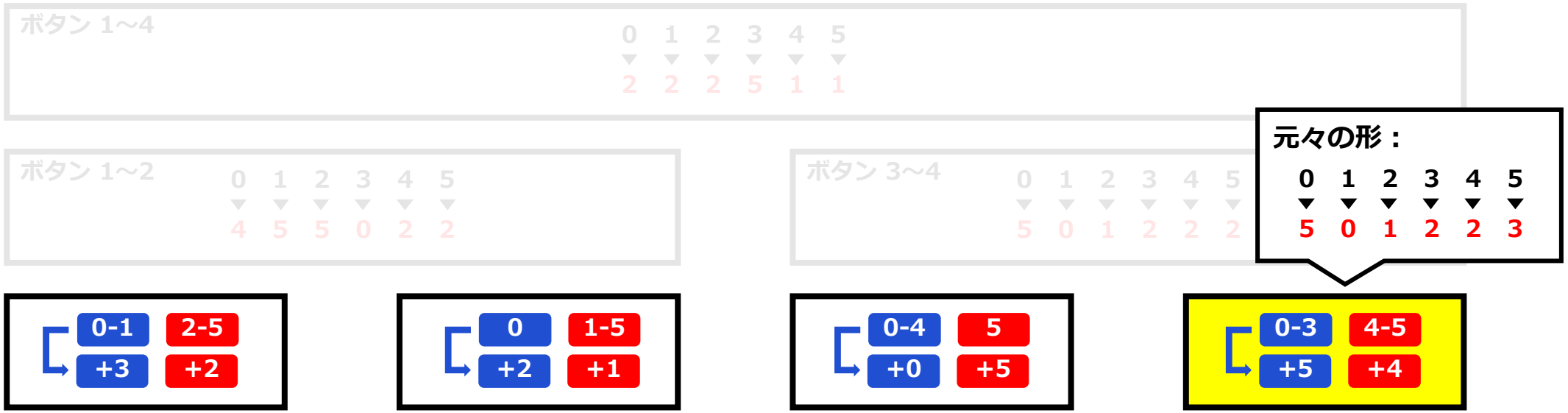
これを圧縮すると、このように変化する

5 セグ木を圧縮：例2



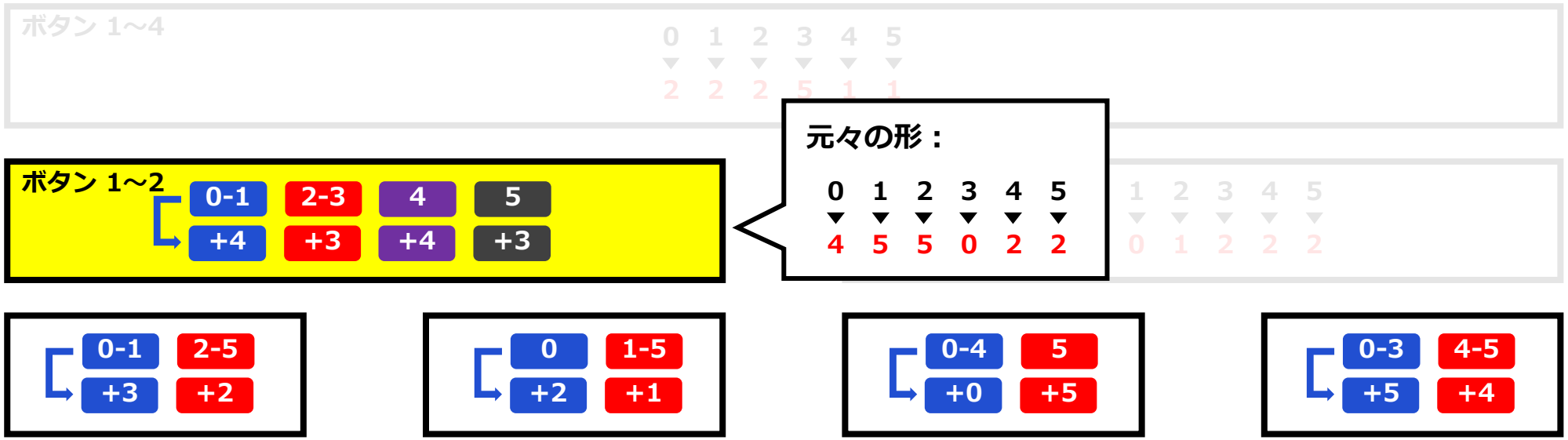
これを圧縮すると、このように変化する

5 セグ木を圧縮：例2



これを圧縮すると、このように変化する

5 セグ木を圧縮：例2

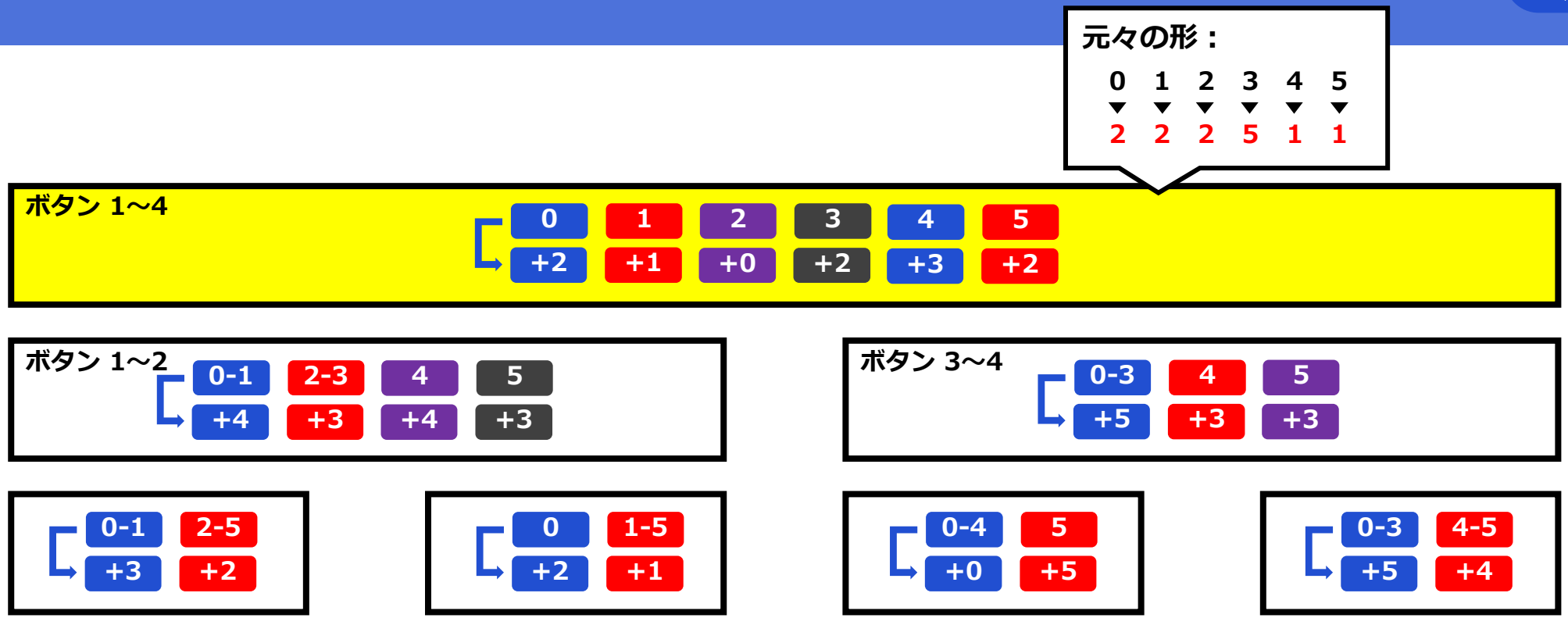


これを圧縮すると、このように変化する

5 セグ木を圧縮：例2

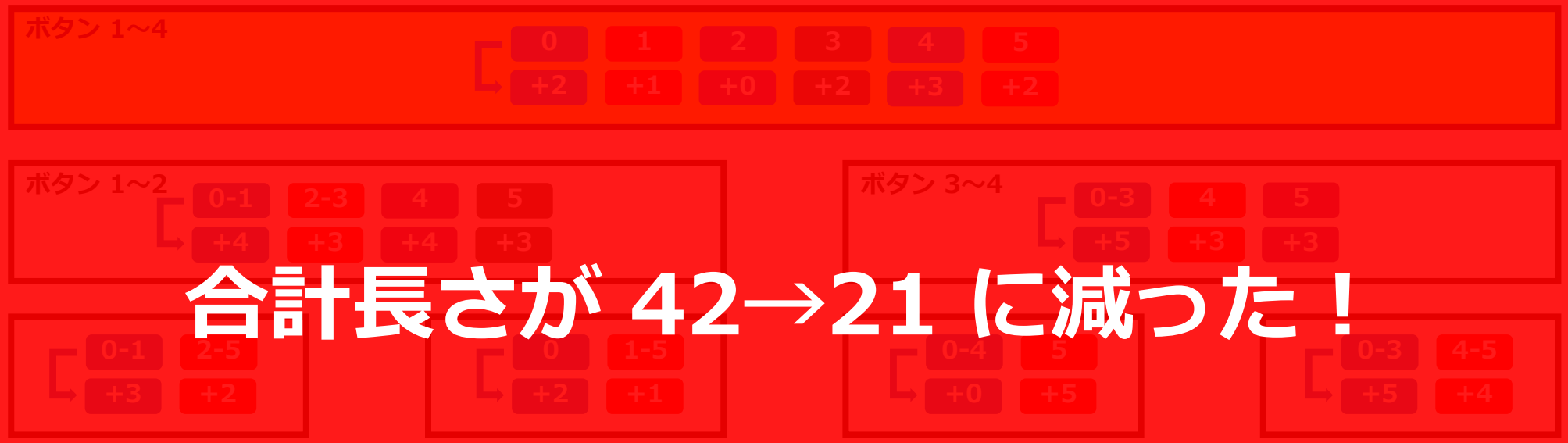
これを圧縮すると、このように変化する

5 セグ木を圧縮：例2



これを圧縮すると、このように変化する

5 セグ木を圧縮：例2



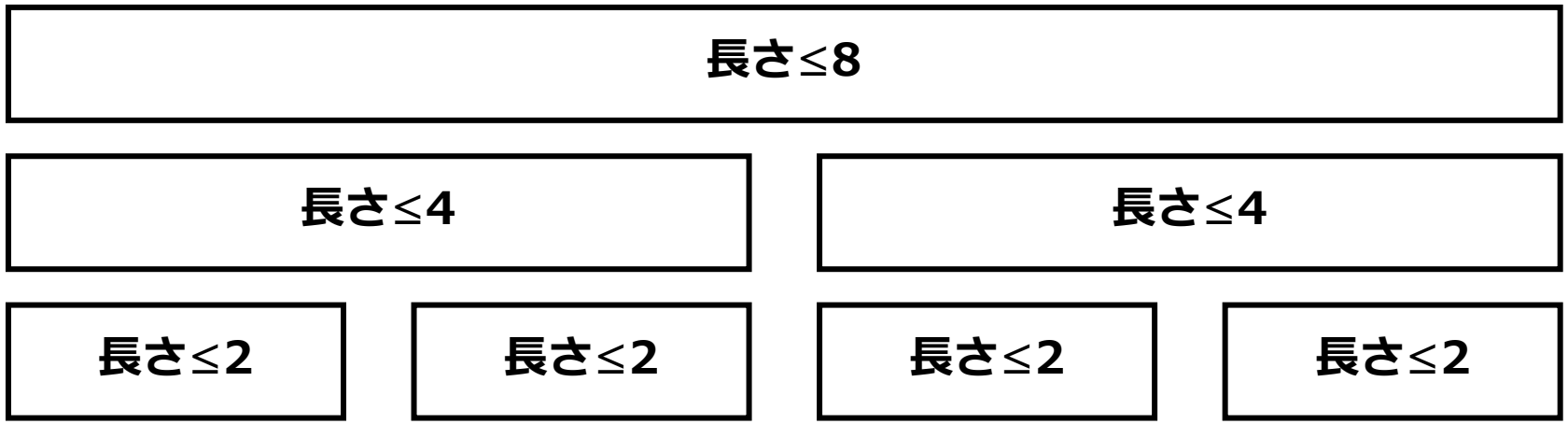
合計長さが 42→21 に減った！

これを圧縮すると、このように変化する

Q. この単純な圧縮方法で
データサイズはどのくらい減るのか？

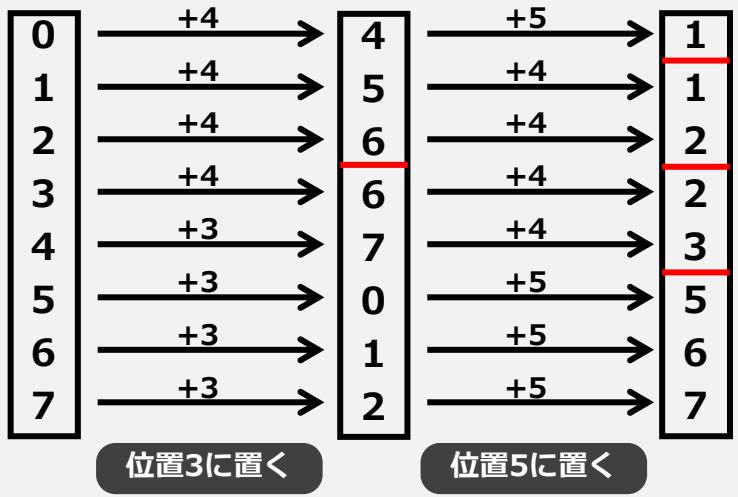
実は、ボタンを k 回押すことに対応するノードは、長さ $2k$ 以内に圧縮可能

例：$M = 4$ の場合は下図の通り



実は、ボタンを k 回押すことに対応するノードは、**長さ $2k$ 以内**に圧縮可能

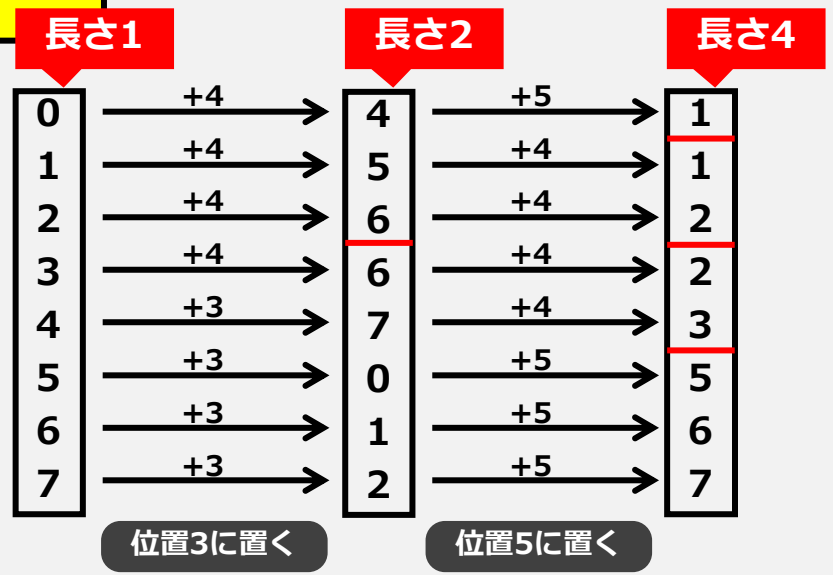
証明



- 右図のように、増分が異なる位置に赤線を引いた図を考える

実は、ボタンを k 回押すことに対応するノードは、長さ $2k$ 以内に圧縮可能

証明



- 右図のように、増分が異なる位置に赤線を引いた図を考える
- このとき、圧縮した後の長さは (赤線の数) + 1

実は、ボタンを k 回押すことに対応するノードは、長さ $2k$ 以内に圧縮可能

証明

- 右図のように、増分が異なる位置に赤線を引いた図を考える
- このとき、圧縮した後の長さは (赤線の数) + 1

1 回の操作を行うと、赤線はどこで追加されるのか？

- “操作が与える増分” が変わるところ
- 位置 p に置いたとき、変わるところは「0 の直前」と「p の直前」の 2 箇所だけ

実は、ボタンを k 回押すことに対応するノードは、長さ $2k$ 以内に圧縮可能

つまり、ボタン 1 回で赤線は 2 個しか増えない

▼

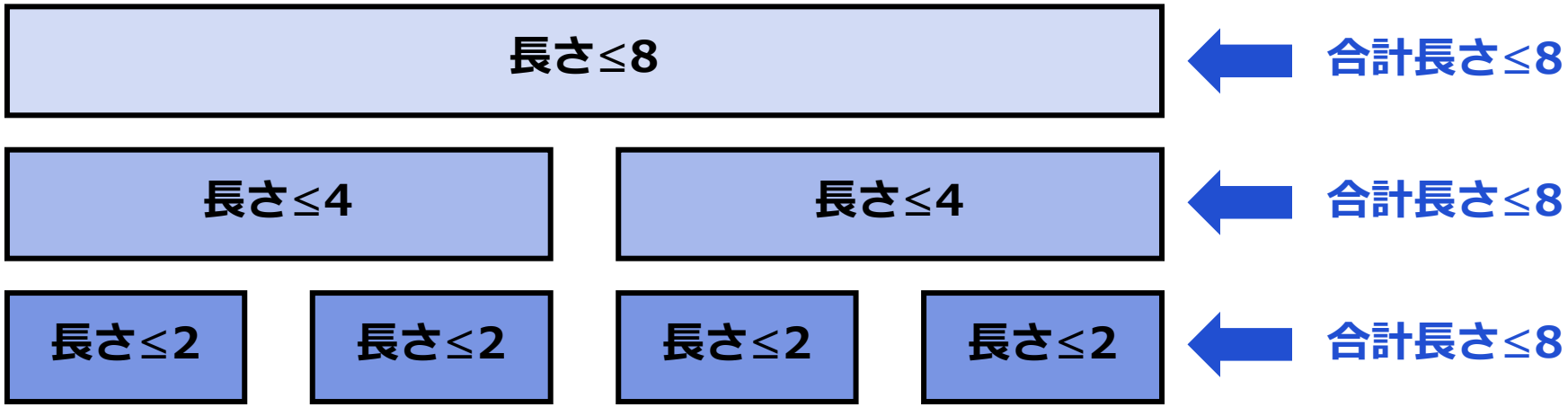
ボタン k 回のノードは、圧縮後の長さが $2k$ 以下！

したがって、セグメント木の各深さごとの長さの合計は $2M$ 以下になる

→合計長さは $2M \log M$ 以下

したがって、セグメント木の各深さごとの長さの合計は $2M$ 以下になる

→合計長さは $2M \log M$ 以下 (下図は $M = 4$ の場合)



5 小課題5 (累計62点)

このように、セグメント木を圧縮すれば小課題 5 が解ける

プログラムの計算量は以下の通り：

- セグメント木の構成：二分探索が必要なので、計算量は $O(M \log^2 M)$
- クエリ処理：これも二分探索が必要なので、計算量は $O(\log^2 M)$
- したがって、全体の計算量は $O((M+Q) \log^2 M)$

62点

※二分探索の処理は、主に「位置 x が圧縮されたノードのどこに存在するか」を調べるために使う

小課題 6

$A_i \leq 20$ または $A_i > N - 20$

6 小課題6でやるべきこと

まず、この問題では以下の 2 つのフェーズを解く必要がある

※小課題 3 の解説を参照

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

6 小課題6でやるべきこと

まず、この問題では以下の 2 つのフェーズを解く必要がある

※小課題 3 の解説を参照

フェーズ1　左の赤・右の青の範囲が徐々に広がっていくフェーズ

フェーズ2　左の赤と右の青がぶつかった後のフェーズ

➡ 実は、フェーズ 2 は小課題 5 で解けている

なのでフェーズ 1 だけを考えればよい

6 小課題6でやるべきこと

どうやってフェーズ 1 を解くか？

6 小課題3の解法（再掲）

以下の変数を用意する

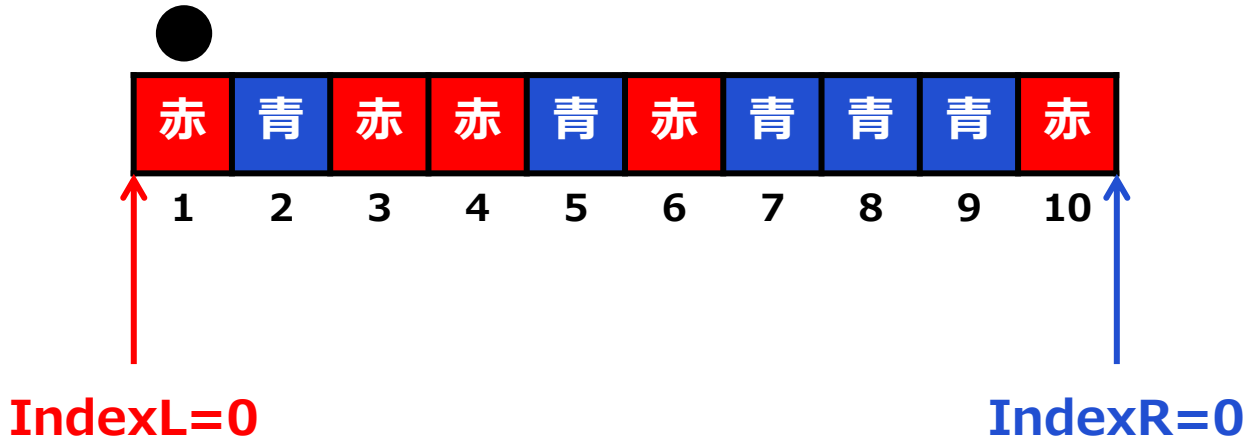
- IndexL：左から何番目までの青タイルが赤になったか
- IndexR：右から何番目までの赤タイルが青になったか

ボールが左から x_L 番目・右から x_R 番目の位置に置かれた場合…

- ボールが左から出るとき：IndexL に x_L を加算
- ボールが右から出るとき：IndexR に x_R を加算

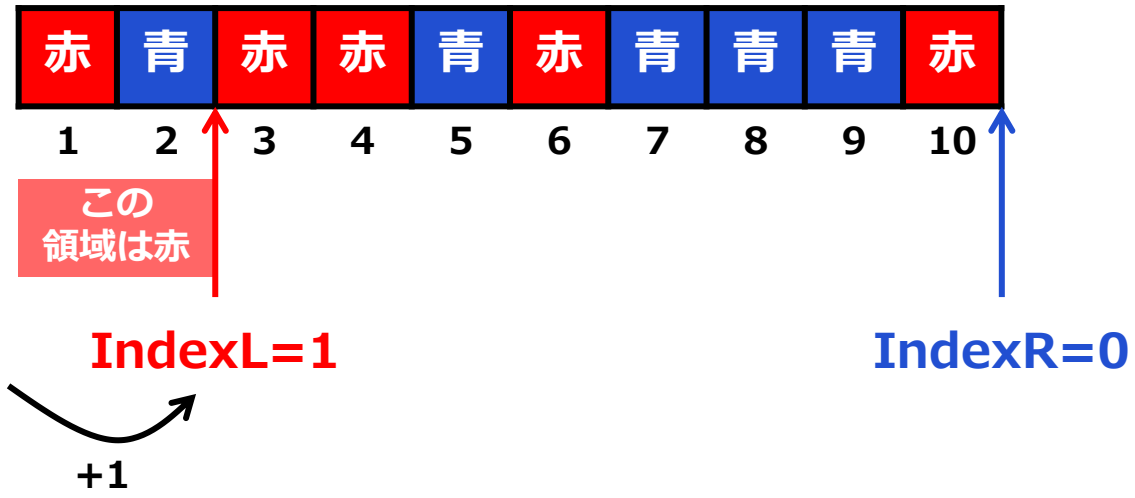
6 小課題3の解法：例

操作	ボールが位置 1 に置かれる



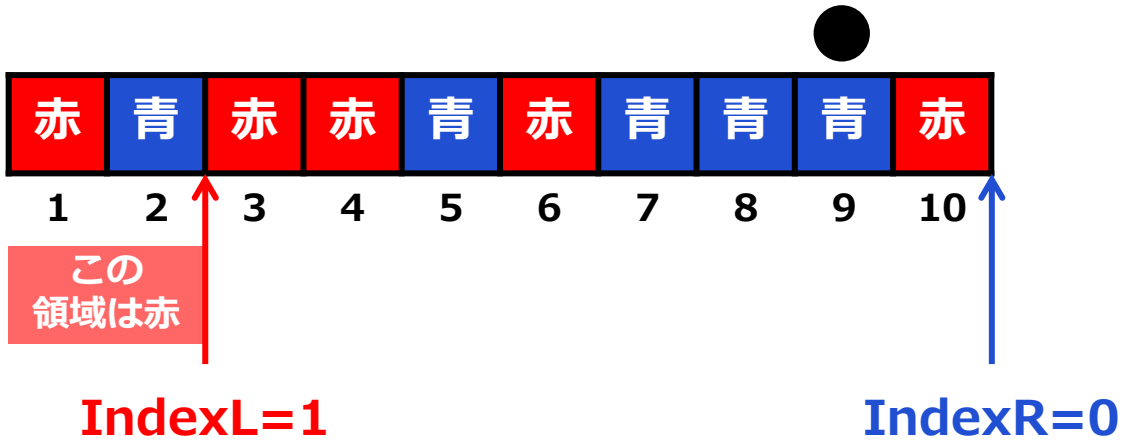
6 小課題3の解法：例

操作	ボールが位置 1 に置かれる



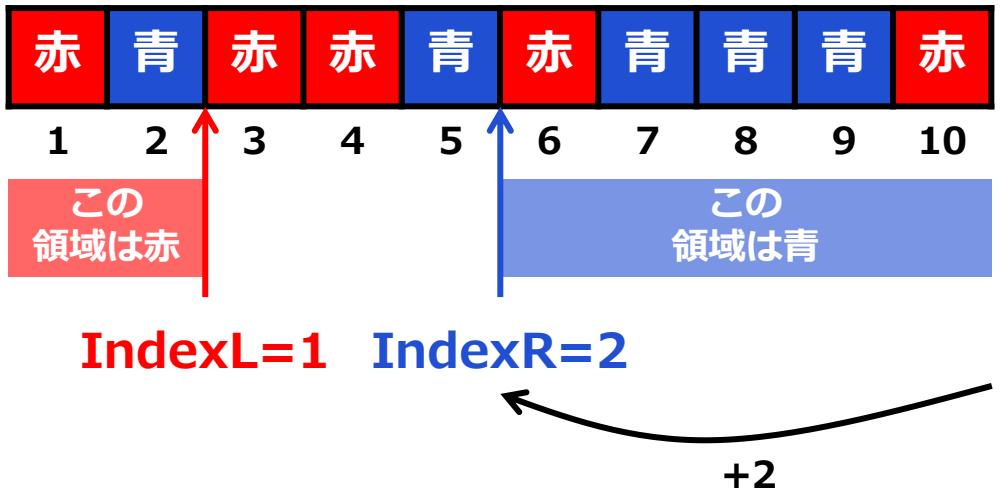
6 小課題3の解法：例

操作	ボールが位置 9 (右から 2 番目) に置かれる



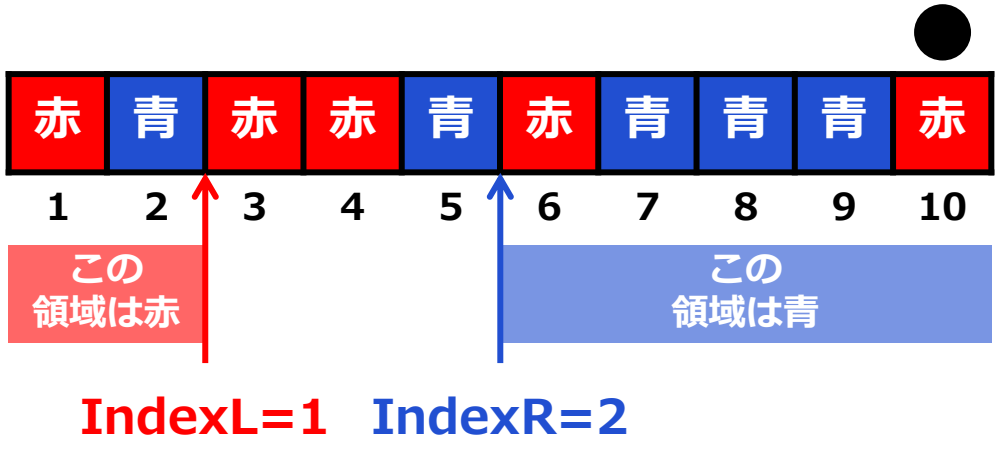
6 小課題3の解法：例

操作	ボールが位置 9 (右から 2 番目) に置かれる



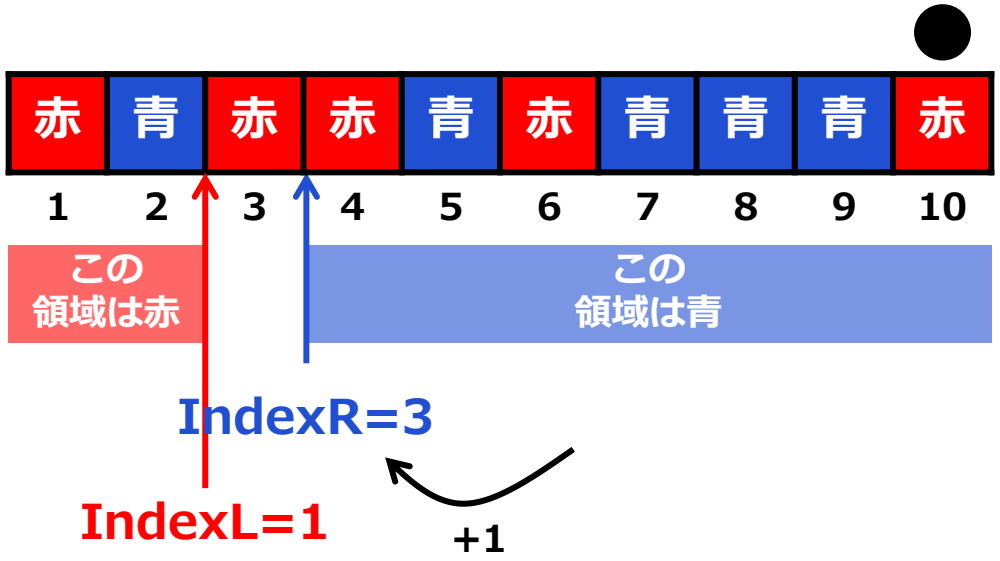
6 小課題3の解法：例

操作	ボールが位置 10 (右から 1 番目) に置かれる



6 小課題3の解法：例

操作	ボールが位置 10 (右から 1 番目) に置かれる



6 小課題3の解法：問題点

しかし、この解法ではボタン当たり $O(1)$

全体で計算量 $O(M)$ かかってしまう

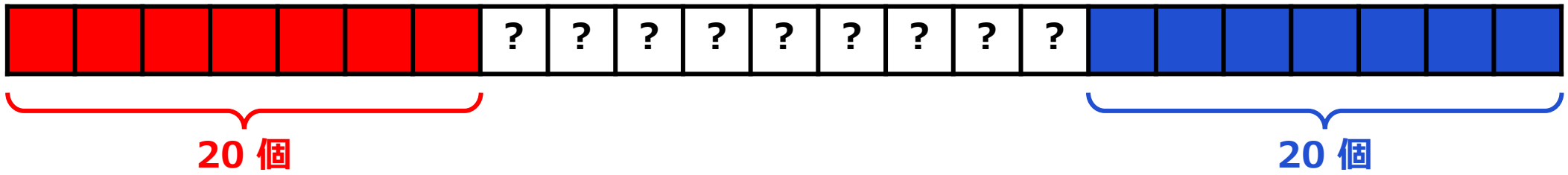
▼

どうにかしていくつかのボタンをまとめて処理できないか？

6 フェーズ1の高速化

高速化の方法

6 フェーズ1の高速化



手始めに、タイルの初期状態が

左端 20 個が赤・右端 20 個が青のときを考えよう

6 フェーズ1の高速化

重要 このとき、ボールを置いた後の動きとしては以下の 2 つがあり得る

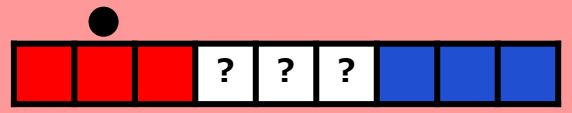
$x_L \leq 20$ の場合

1

$x_R \leq 20$ の場合

2

基本的にはボールが左から出るので
IndexL が x_L 増加



※左の赤と右の青がぶつかったときのみ、ボールが右から出ることがある

6 フェーズ1の高速化

重要 このとき、ボールを置いた後の動きとしては以下の 2 つがあり得る

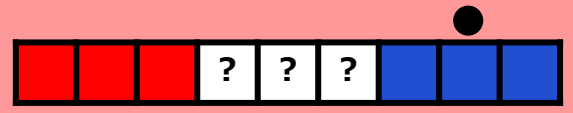
$x_L \leq 20$ の場合

1

$x_R \leq 20$ の場合

2

基本的にはボールが右から出るので
IndexR が $x_R - 1$ 増加



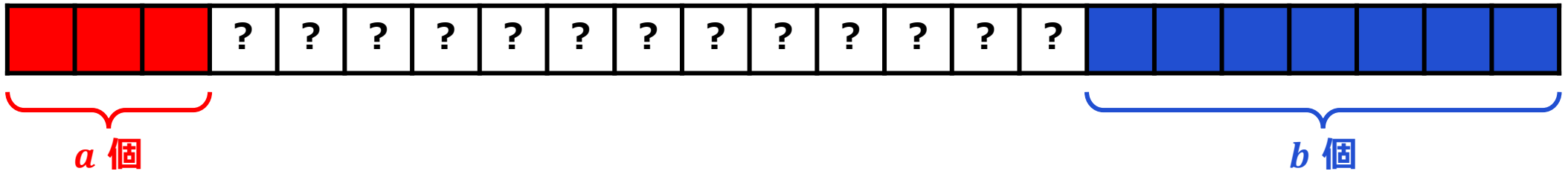
※x_R ではなく $x_R - 1$ になる理由は、ボールが動く前に、ボールの場所のタイルが青から赤に変わるため

6 フェーズ1の高速化

したがって、ボタン $l, l+1, \ldots, r$ を押した後の IndexL, IndexR の値は累積和によって計算できる！

➡ r で二分探索をすれば、左の青と右の赤がいつぶつかるかも $O(\log M)$ で計算できる

6 フェーズ1の高速化



次に一般の場合を考えよう

現時点で、タイルの左端 a 個が赤、右端 b 個が青であるとする

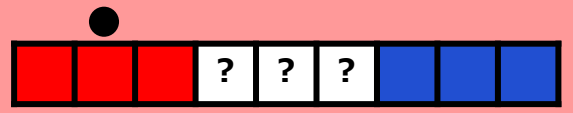
6 フェーズ1の高速化

重要 このとき、ボールを置いた後の動きとしては以下の 3 つがあり得る

$x_L \leq a$ の場合　$x_R \leq b$ の場合　どちらでもない

1　2　3

基本的にはボールが左から出るので
IndexL が x_L 増加



※左の赤と右の青がぶつかったときのみ、ボールが右から出ることがある

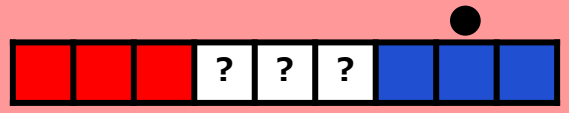
6 フェーズ1の高速化

重要 このとき、ボールを置いた後の動きとしては以下の 3 つがあり得る

$x_L \leq a$ の場合　　$x_R \leq b$ の場合　　どちらでもない

1　　2　　3

基本的にはボールが左から出るので
IndexR が $x_R - 1$ 増加



※x_R ではなく $x_R - 1$ になる理由は、ボールが動く前に、ボールの場所のタイルが青から赤に変わるため

6 フェーズ1の高速化

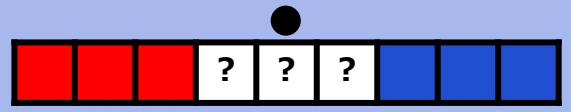
重要 このとき、ボールを置いた後の動きとしては以下の 3 つがあり得る

$x_L \leq a$ の場合 → 1

$x_R \leq b$ の場合 → 2

どちらでもない → 3

どう動くかは分からない…



ここで、どちらでもない場合は

計算量 $O(1)$ かけて処理する必要がある

6 フェーズ 1 の高速化

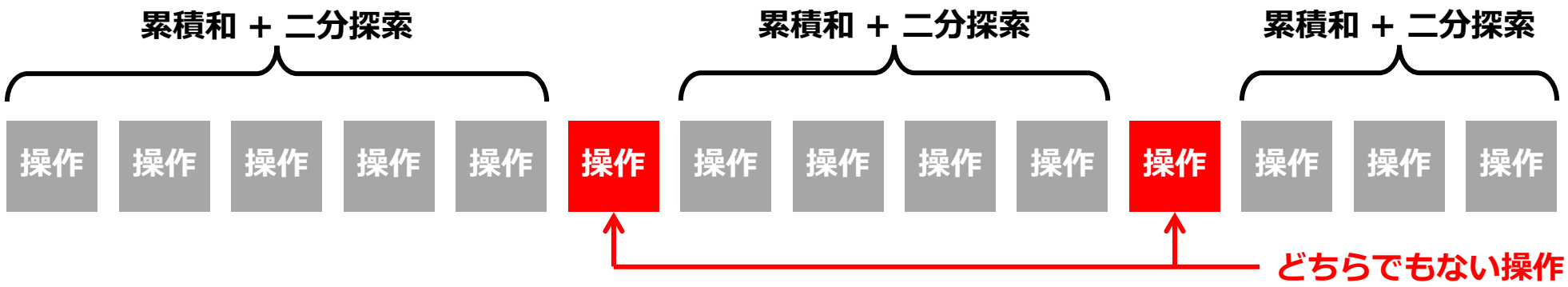
しかし、この小課題では $A_i \leq 20$ または $A_i > N - 20$ なので

“どちらでもない” になる可能性があるのは…

1. $1 \leq x_L \leq 20$ となる操作のうち最初の 20 個
2. $2 \leq x_R \leq 20$ となる操作のうち最初の 20 個
3. $x_R = 1$ となる操作のうち最初の 1 個

合計 41個

6 小課題6 (累計79点)



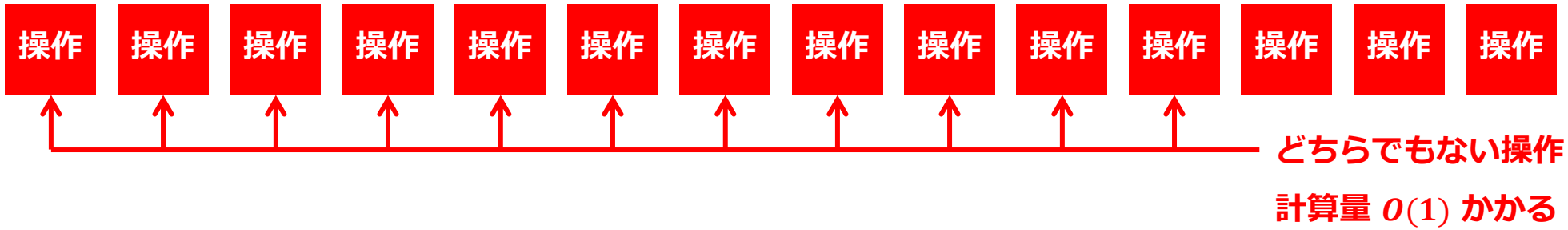
したがって、「飛ばせる部分」は累積和 + 二分探索を使い

どうしようもない部分は計算量 $O(1)$ で処理すれば、計算回数は約 $40 \log M$ に

小課題 7

追加の制約はない

7 満点解法のアイデア



小課題 6 の解法を一般の制約に適用させると

「どちらでもない操作」を最大 M 回行うことになると思うかもしれない

7 満点解法のアイデア

しかし実際は、どちらでもない操作は $O(\log N)$ 回しか行われない

7 満点解法のアイデア

しかし実際は、どちらでもない操作は $O(\log N)$ 回しか行われない

なぜか？ 左の赤が a 個、右の青が b 個の状態で「どうしようもない操作」を行うとき：

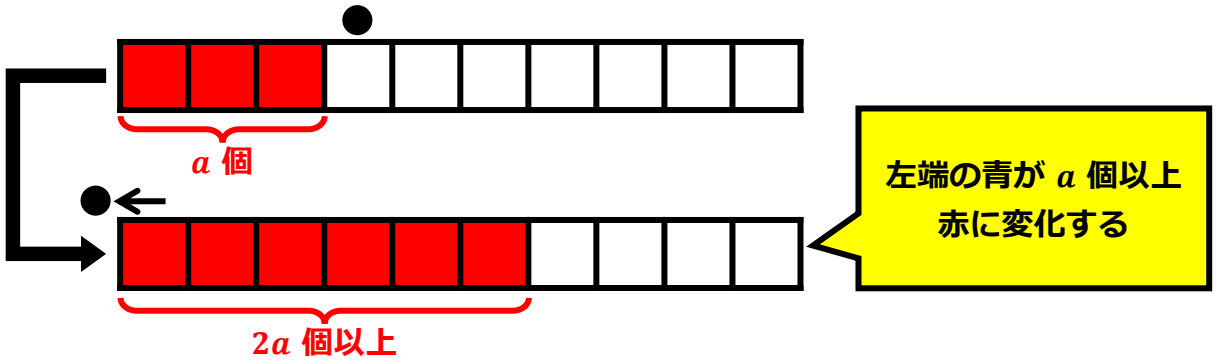
- ボールが左から出るとき、操作後の左の赤は $2a$ 個以上に
- ボールが右から出るとき、操作後の右の青は $2b$ 個以上に

7 満点解法のアイデア

しかし実際は、どうしようもない操作は $O(\log N)$ 回しか行われない

なぜか？ 左の赤が a 個、右の青が b 個の状態で「どうしようもない操作」を行うとき：

- ボールが左から出るとき、操作後の左の赤は $2a$ 個以上に
- ボールが右から出るとき、操作後の右の青は $2b$ 個以上に

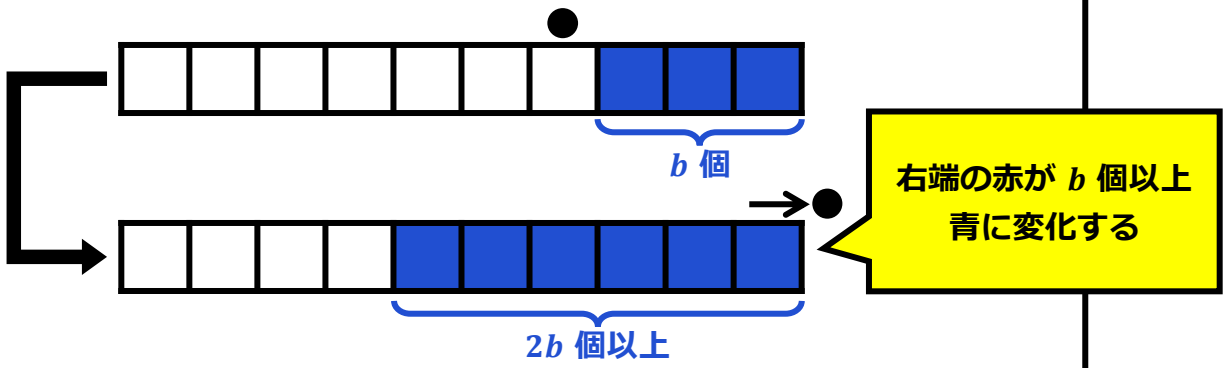


7 満点解法のアイデア

しかし実際は、どうしようもない操作は $O(\log N)$ 回しか行われない

なぜか？ 左の赤が a 個、右の青が b 個の状態で「どうしようもない操作」を行うとき：

- ボールが左から出るとき、操作後の左の赤は $2a$ 個以上に
- **ボールが右から出るとき、操作後の右の青は $2b$ 個以上に**



7 満点解法のアイデア

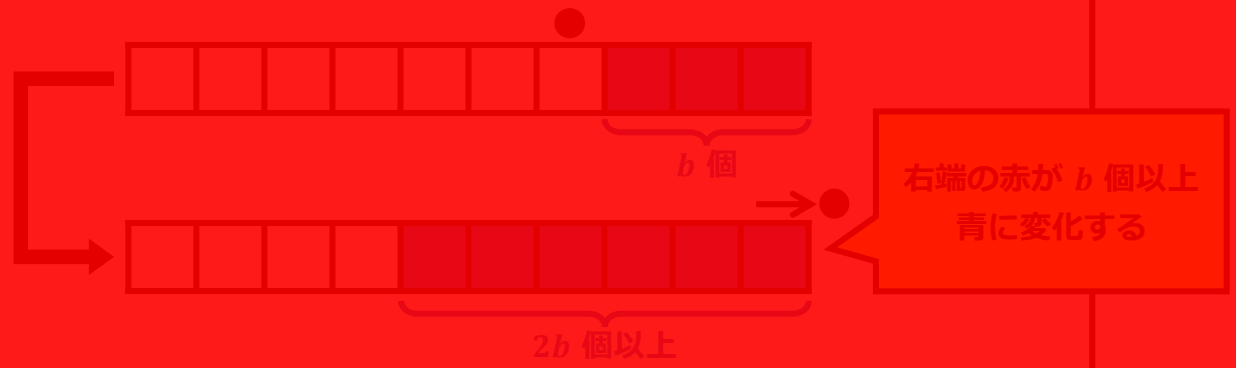
しかし実際は、どうしようもない操作は $O(\log N)$ 回しか行われない

なぜか？

左の赤が a 個、右の青が b 個の状態で「どうしようもない操作」を行うとき：

- ボールが左から出るとき、操作後の左の赤は $2a$ 個以上に
- ボールが右から出るとき、操作後の右の青は $2b$ 個以上に

小課題 6 と同じ方針で解けそう！



7 満点解法の本題

だが、まだ解けたわけではない

次の「どちらでもない操作」がいつ来るかを
高速に求める必要がある！

7 満点解法の本題

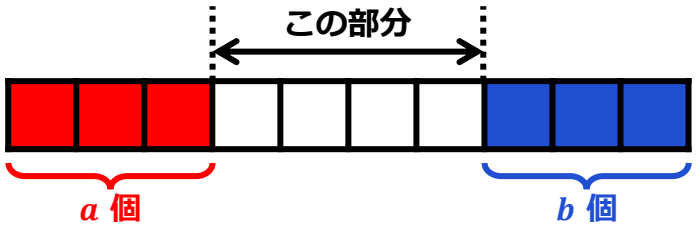
次の「どちらでもない操作」は

どうやって求めるか？

7 “どちらでもない操作” をどう求めるか？

まず、現時点で左の赤が a 個、右の青が b 個であり、ボタン x までの処理が終わっている場合…

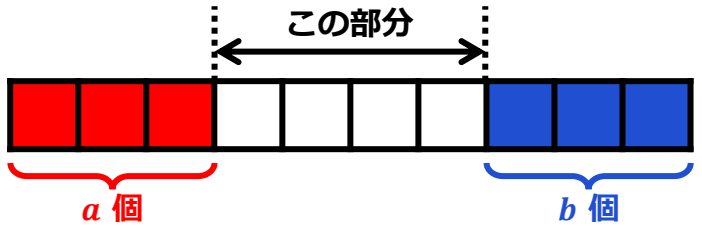
- 次のどちらでもない操作の番号は $a+1 \leq A_i \leq N-b$ を満たす最小の $i\ (>x)$



7 “どちらでもない操作” をどう求めるか？

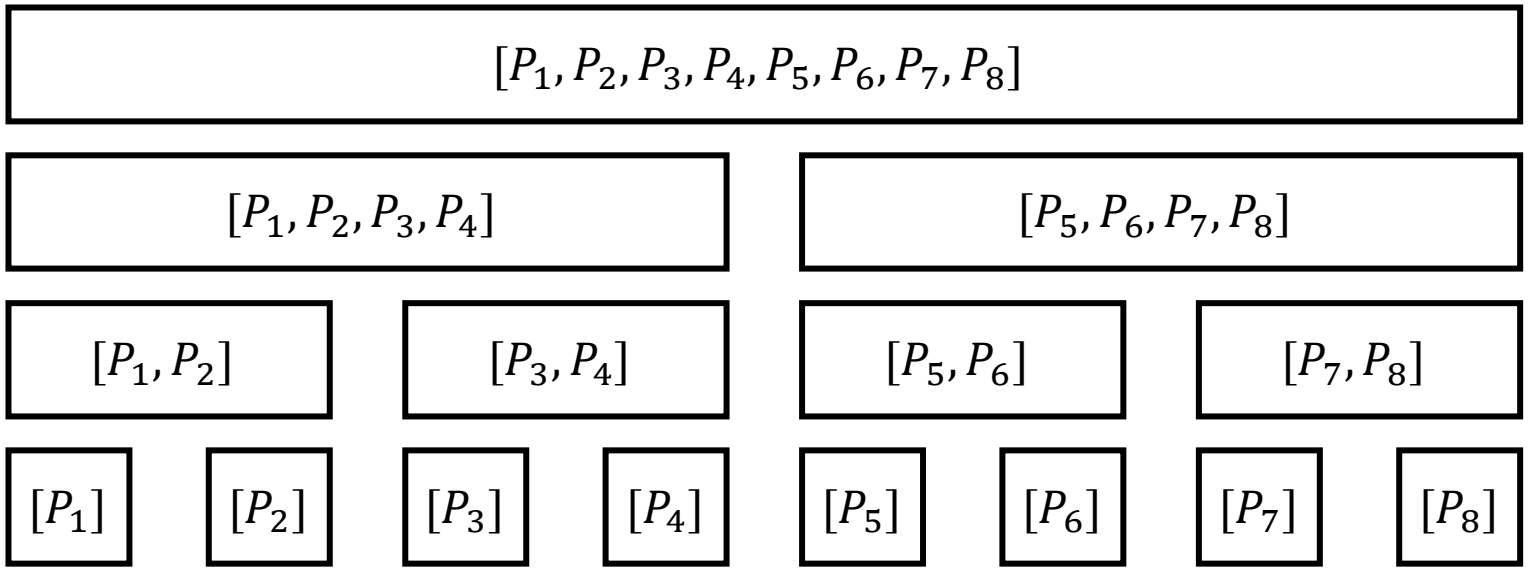
まず、現時点で左の赤が a 個、右の青が b 個であり、ボタン x までの処理が終わっている場合…

- 次のどちらでもない操作の番号は $a+1 \leq A_i \leq N-b$ を満たす最小の $i\ (>x)$



➡ これは、領域木というデータ構造を使えば高速

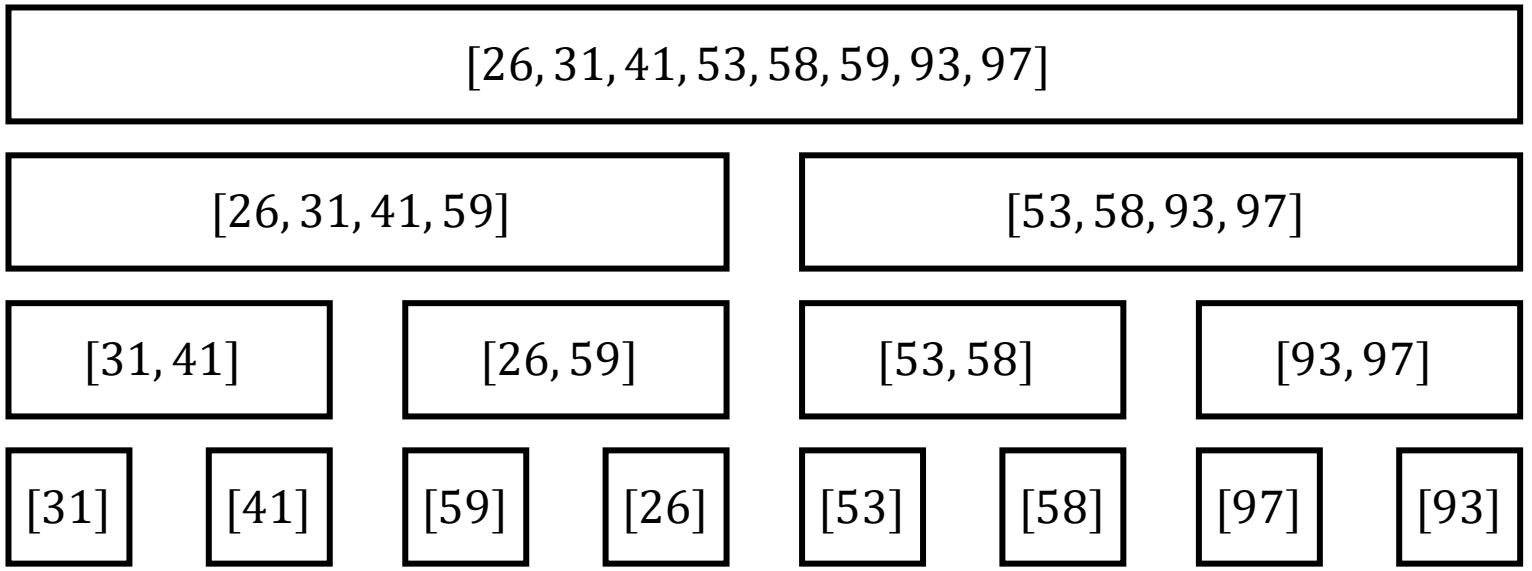
領域木とは



セグメント木の一種

区間 $[L, R]$ に対応するノードに $[P_L, P_{L+1}, \ldots, P_R]$ をソートして格納

7 領域木とは



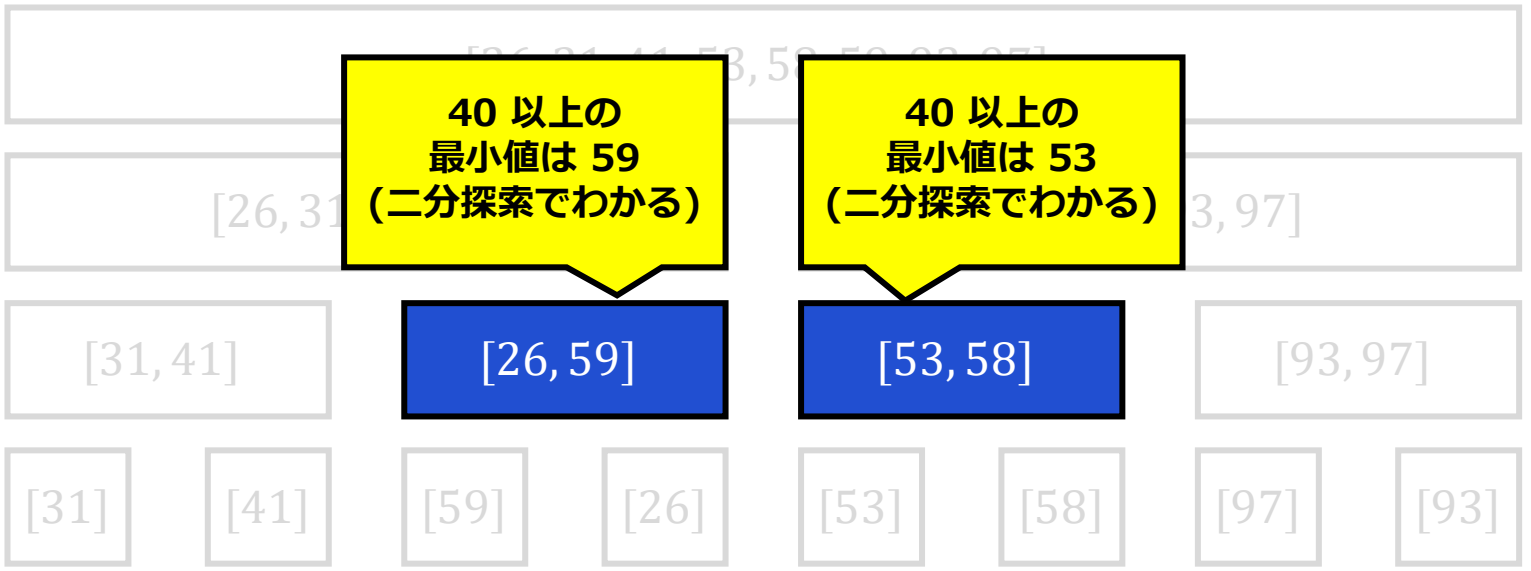
たとえば $X = [31, 41, 59, 26, 53, 58, 97, 93]$ の場合、領域木は上図の通り

※この例では 1 個の葉ノードに 1 つの要素しか存在しないが、0 個や 2 つ以上に拡張することもできる

領域木上で二分探索をすると

$P_i \geq x\ (l \leq i \leq r)$ となる最小の P_i がわかる

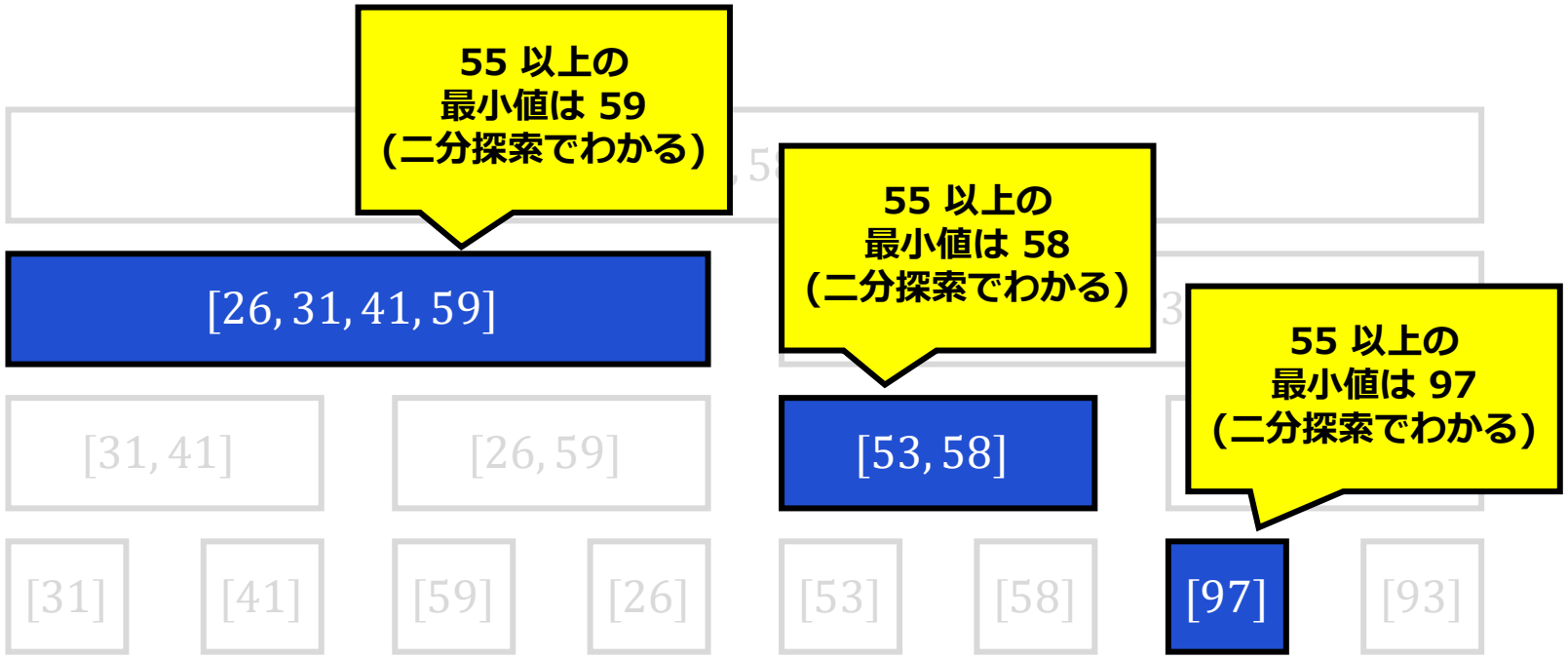
領域木でクエリ処理



たとえば $P_i \geq 40\ (3 \leq i \leq 6)$ となる最小の P_i はいくつか？

➡ 59 と 53 のうち小さい方「53」

領域木でクエリ処理



たとえば $P_i \geq 55\ (1 \leq i \leq 7)$ となる最小の P_i はいくつか？

➡ 59, 58, 97 のうち小さい方「58」

領域木でクエリ処理

このように領域木を使うと、次の「どちらでもない操作」を

計算量 $O(\log N \log M)$ で求めることができる

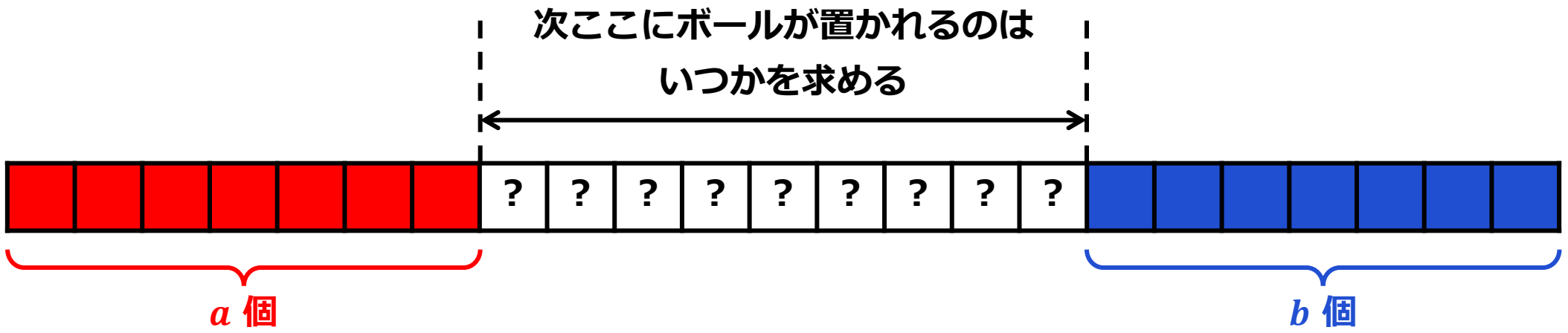
▼

しかし、「どちらでもない操作」は全体を通して $O(Q \log N)$ 回

log が 3 つとなり、間に合うかは微妙…

計算量を改善するには？

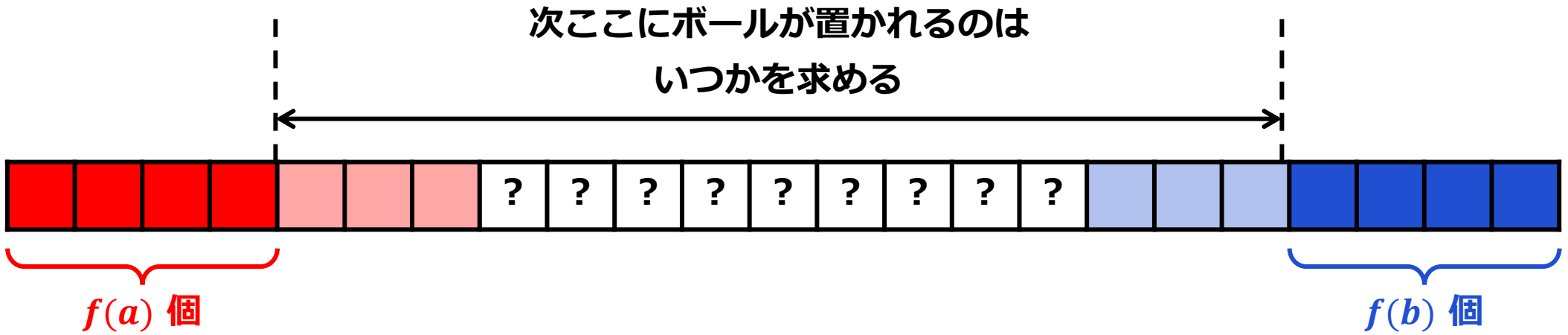
7 計算量の改善：アイデア



元々の方法では、$a+1 \leq A_i \leq N-b$ を満たす最小の i を

「次のどちらでもない操作」にしていた

7 計算量の改善：アイデア



しかし、n 以下で最大の 2^k 数を $f(n)$ とするとき

$f(a)+1 \leq A_i \leq N-f(b)$ を満たす i を

「次のどちらでもない操作」にすると上手くいく

なぜ上手くいくか？

7 2 べきが上手くいく理由 (1/2)

- $f(a), f(b)$ の組は高々 $\log^2 N$ 個しかない
- したがって、すべての $f(a), f(b), x$ に対して「$f(a) + 1 \leq A_i \leq N - f(b)$ を満たす最小の $i\ (\geq x)$」を前計算できる※

▼

計算量 $O(1)$ で「次のどちらでもない操作」がわかる

※前計算は累積 max のようなことをやると計算量 $O(N \log^2 N)$ でできる [メモリ量に比例する時間]

7 2 べきが上手くいく理由 (2/2)

さらに、範囲を $f(a)+1$ 以上 $f(b)$ 以下に緩和させても

どちらでもない操作が起こる回数が $O(\log N)$ 回であることに変わりはない

なぜか？ 一度「どうしようもない操作」が起こると

$f(a), f(b)$ のうち少なくとも一方が 2 倍以上になるから

7 小課題7 (累計100点)

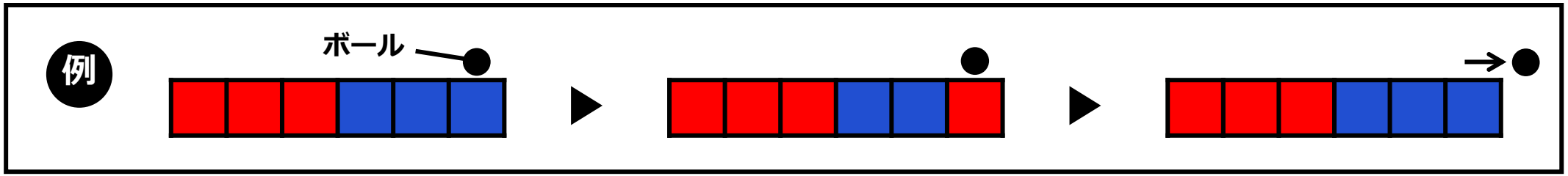
- このように 2 べきごとに累積 max を記録する方針を使うと、各クエリ $O(\log N\,(\log N + \log M))$ で答えを出すことができる

- $N, M, Q \leq 120000$ なので、実行時間制限の 2.5 秒には間に合うはず

実装上の注意点

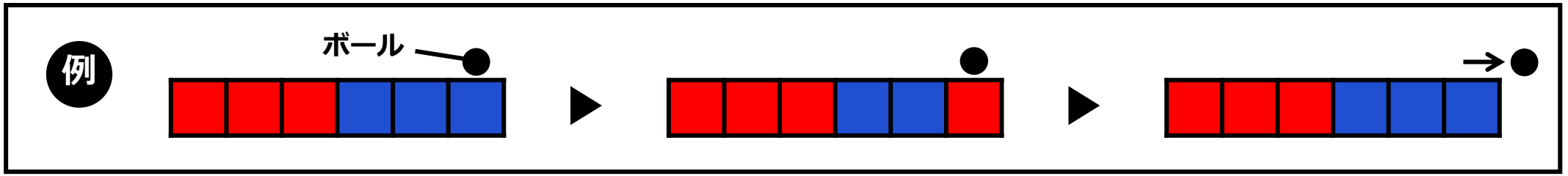
7 注意点

ボールがタイル N に置かれた場合、盤面が変わらないことがある



7 注意点

ボールがタイル N に置かれた場合、盤面が変わらないことがある



上手い実装をしなければ、WA や TLE になる！

たとえば、最初に「左の赤が $a = 0$ 個」「右の青が $b = 0$ 個」と設定した場合…
ボールが毎回タイル N に置かれたとき、全部「どちらでもない操作」になってしまう

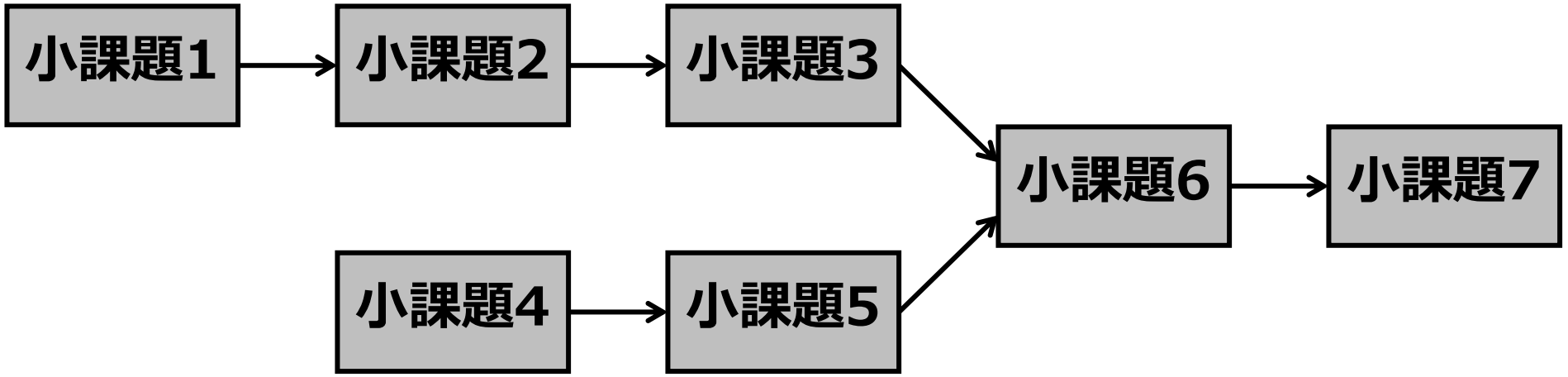
解法のまとめ

解法のまとめ

小課題1	愚直にシミュレーション
小課題2	位置 x に置いて左から出た場合、左から x 個の青が赤になることに気づく
小課題3	一度 RRRRBBBB の形になったら $\bmod (N+1)$ で計算できることに気づく
小課題4	圧縮無しのセグメント木
小課題5	圧縮ありのセグメント木
小課題6	累積和 + 二分探索で「左の赤と右の青」がいつぶつかるのかを求める この制約だと、どうしようもない操作※は高々 40 回以内
小課題7	どちらでもない操作が $O(\log N)$ 回以内であることに気づく + 領域木の代わりに「2べきのデータ構造」を使うと log が 2 つになる

※どうしようもない操作は、左の赤にも右の青にも含まれない場所にボールを置く操作

解法のまとめ

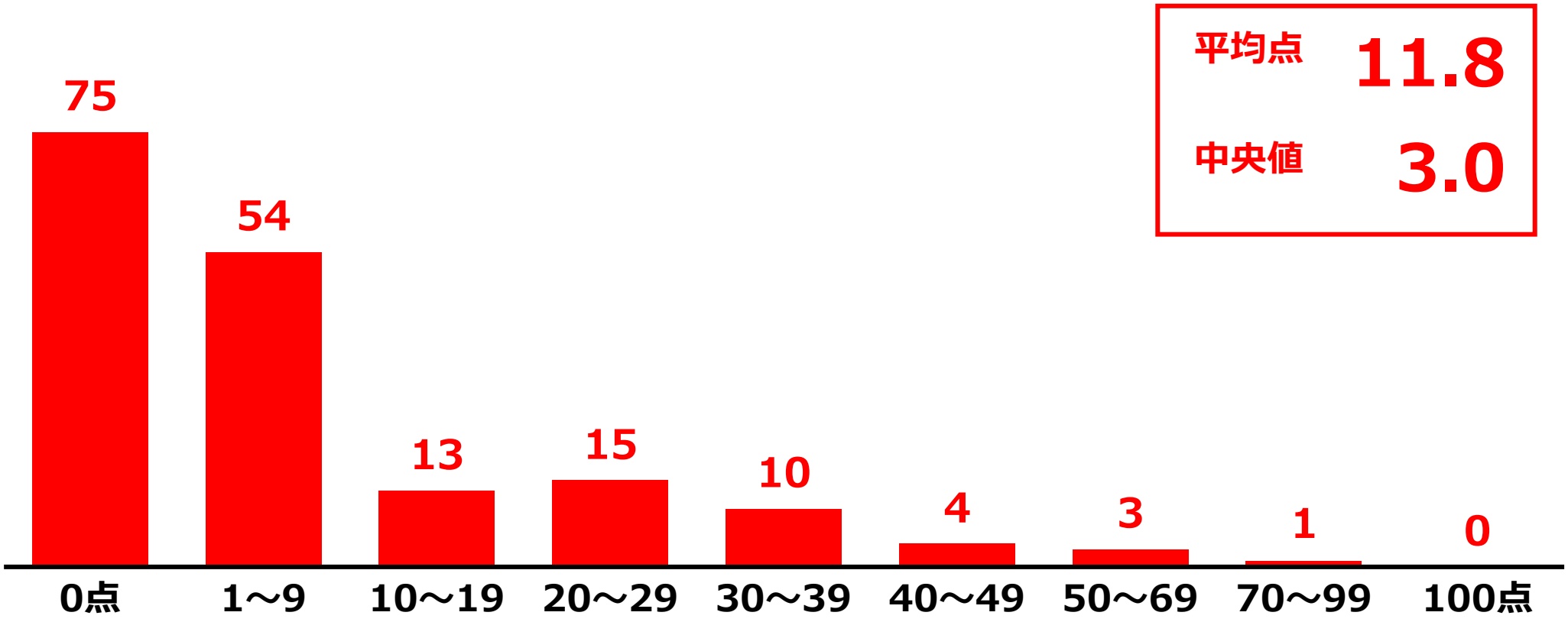


本問題は満点までのステップ数が多いが、誘導も多い

JOI では、小課題の誘導に乗ることが重要になる場合もある

得点分布

191 / 191


平均点 11.8
中央値 3.0
75
54
13
15
10
4
3
1
0
0点
1～9
10～19
20～29
30～39
40～49
50～69
70～99
100点