に・ぜろ・に・よん

パソコン甲子園2024

全国高等学校パソコンコンクール
プログラミング部門 本選問題解説

問題セット

1 宝石

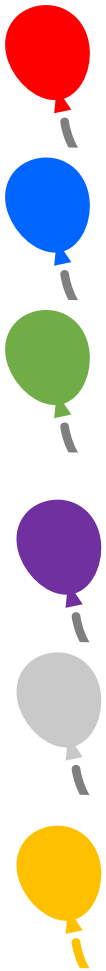
2 2024を法として合同

3 名手が放つ矢

4 荷物渡し1

5 光線の反射

6 荷物の配置パターン



7 盆栽育成ゲーム

8 多角形の重なり

9 荷物渡し2

10 学習データセットの分析

11 六角沼からの脱出

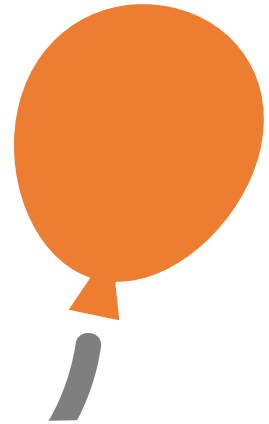
12 倉庫管理ロボット

問題セット

#	タイトル	分野	得点	難易度	
				思考	実装
1	**宝石**	基礎	2	☆	☆
2	**2024を法として合同**	基礎	3	★	☆
3	**名手が放つ矢**	探索	4	★	★
4	**荷物渡し 1**	シミュレーション	5	★	★☆
5	**光線の反射**	整数論	8	★★☆	★
6	**荷物の配置パターン**	数え上げ（動的計画法）	10	★★★	★★
7	**盆栽育成ゲーム**	組み合わせ（動的計画法）	10	★★★	★★
8	**多角形の重なり**	計算幾何・データ構造	10	★★★	★★★
9	**荷物渡し 2**	データ構造・シミュレーション	12	★★★	★★★☆
10	**学習データセットの分析**	しゃくとり法・累積和・二分探索	12	★★★☆	★★★☆
11	**六角沼からの脱出**	累積和・グラフ・二分探索	12	★★★☆	★★★★
12	**倉庫管理ロボット**	データ構造	12	★★★★	★★★★☆

問題１　宝石　　（２点）

正答数:　　31 チーム



問題１　宝石

概要

- 宝石がN個ある。宝石はM種類ある。
- それぞれの種類で、最も個数が少ない宝石と最も個数が多い宝石の個数差は高々1である。
- 最も個数が少ない種類の個数と最も個数が多い種類の個数を求めよ。
- $2 \leq N \leq 50, 2 \leq M \leq N$

問題１　宝石

入出力例

$N = 7$
$M = 3$

r r e e s s s

2　2　3

$\text{floor}(\frac{7}{3}) = 2$

$\text{ceiling}(\frac{7}{3}) = 3$

$N = 8$
$M = 3$

r r e e e s s s

2　3　3

$\text{floor}\left(\frac{8}{3}\right) = 2$

$\text{ceiling}\left(\frac{8}{3}\right) = 3$

$N = 9$
$M = 3$

r r r e e e s s s

3　3　3

$\text{floor}\left(\frac{9}{3}\right) = 3$

$\text{ceiling}\left(\frac{9}{3}\right) = 3$

解法

- 最も数が少ない宝石(ルビー)の数

  $\text{floor}\left(\frac{N}{M}\right) = \text{N/M}$

- 最も数が多い宝石(サファイア)の数

  $\text{ceiling}\left(\frac{\text{N}}{\text{M}}\right) = \text{(N+M-1)/M}$

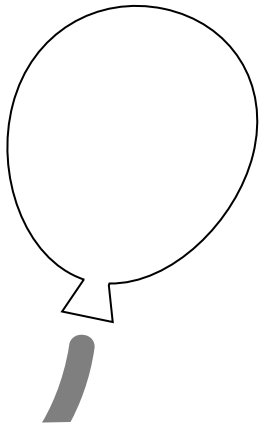
問題１　宝石

解答例（C++）

```
#include<iostream>

int main(){
  int N, M;
  std::cin >> N >> M;
  std::cout << N/M << " " << (N+M-1)/M << std::endl;
  return 0;
}
```

問題２　2024を法として合同　（３点）

正答数:　　31チーム



問題２　2024を法として合同

概要

- ２つの整数の差が自然数nの倍数のとき、それらはnを法として合同という。
- 整数xが与えられたとき、 xと2024を法として合同な、0以上2024未満の整数yを求めよ。

入出力例

- $2025 - 1 = 2024$である。したがって、 1は2025と2024を法として合同な0以上2024未満のただ１つの整数である。
- $-2023 - 1 = -2024$である。したがって、 1は-2023と2024を法として合同な0以上2024未満のただ１つの整数である。

問題２　2024を法として合同

解法１

- 繰り返し処理で (x - y) % 2024 == 0なるyを探す

解答例（C++）

```
#include<iostream>

int main(){
  long long x;
  std::cin >> x;

  for ( long long y = 0; y < 2024; y++ ){
    if ( (x - y) % 2024 == 0 ){
      std::cout << y << std::endl;
      return 0;
    }
  }
}
```

問題２　2024を法として合同

解法２

- 剰余演算で ((x % 2024) + 2024) % 2024 を計算する

剰余演算の結果が 0 でないときの符号は、、、
C++/C, Java ： **被除数**の符号と一致
−3÷2 → −1あまり−1 と考える

Python, Ruby: 除数の符号と一致
−3÷**2** → −2あまり**+1** と考える

C++/C, Javaの場合

$$\text{答え} = \begin{cases} x\%2024 + 2024, & (x\%2024) < 0 \\ x\%2024, & (x\%2024) \geq 0 \end{cases}$$

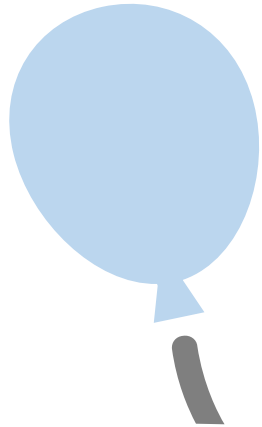
→剰余の性質を利用して場合分けを取り除くと
$$\text{答え} = (x\%2024 + 2024)\%2024$$

解答例（C++）

```
#include<iostream>

int main(){
  long long x;
  std::cin >> x;

  std::cout
      << (x%2024+2024)%2024
      << std::endl;
  return 0;
}
```

問題３　名手が放つ矢　（４点）

正答数:　　31チーム



問題３　名手が放つ矢

概要

- マサムネ君は、弓を引く強さの段階を変えることで、異なる速さで矢を放つことができる。
- 最も弱く弓を引いた0段階から最も強く弓を引いたN段階まで、１段階ずつ強さを選ぶことができ、強く引くほど矢は速く飛ぶ。
- 各段階での矢の速さv_iと、次の段階まで弓を引くのに要する時間t_i、的までの距離Xが与えられる。
- 0　段階の状態から始めて、的に矢が届くまでの最短の時間を求めるプログラムを作成せよ。

制約

- $1 \leq N \leq 100{,}000$
- $1 \leq X \leq 10^9$
- $1 \leq t_i \leq 10^9$
- $1 \leq v_i \leq 10^9$、$v_i < v_{i+1}$

問題３　名手が放つ矢

解法

- 0段階からN段階まで弓を引いたそれぞれの場合について、引き始めてから的に矢が届くまでの時間を求め、その中で最短の時間を出力すればよい。
- 弓を0段階まで引くためにかかる時間s_0は$s_0 = 0$であり、$i\ (i \geq 1)$段階まで引くためにかかる時間s_iは$s_i = s_{i-1} + t_i$と昇順に求められる。
- 的までの距離がXで矢の速さがvのとき、矢を放ってから的に届くまでの時間は、$\frac{X}{v}$である。
- よって、それぞれの場合の引き始めてから的に矢が届くまでの時間a_iは$a_i = s_i + \frac{X}{v_i}$と計算でき、$a_i$の最小値が求めたい答え。

	0	1	2	3	4	5
t	0	1	1	1	1	1
s	0	1	2	3	4	5
v	1	2	3	5	8	13

問題３　名手が放つ矢

注意点

- Xはvで割り切れない場合があるので、実数型で割り算を行う必要がある。
- s_iの計算結果が32-bit整数型の範囲に収まらない入力もあり得るので、64-bit整数型を使用する必要がある。

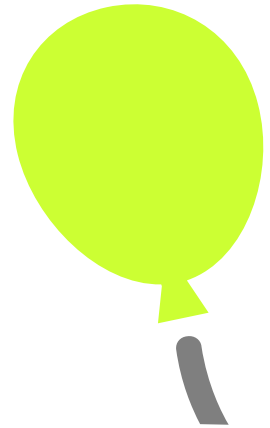
問題３　名手が放つ矢

解答例（C++）

```
long long s = 0;
double ans = double(X) / v[0];
for (int i = 1; i <= N; i++) {
    s += t[i];
    double a = s + double(X) / v[i];
    if (ans > a) ans = a;
}

std::cout.setf(std::ios_base::fixed,std::ios_base::floatfield);
std::cout << res << std::endl;
```

問題４　荷物渡し１　（５点）

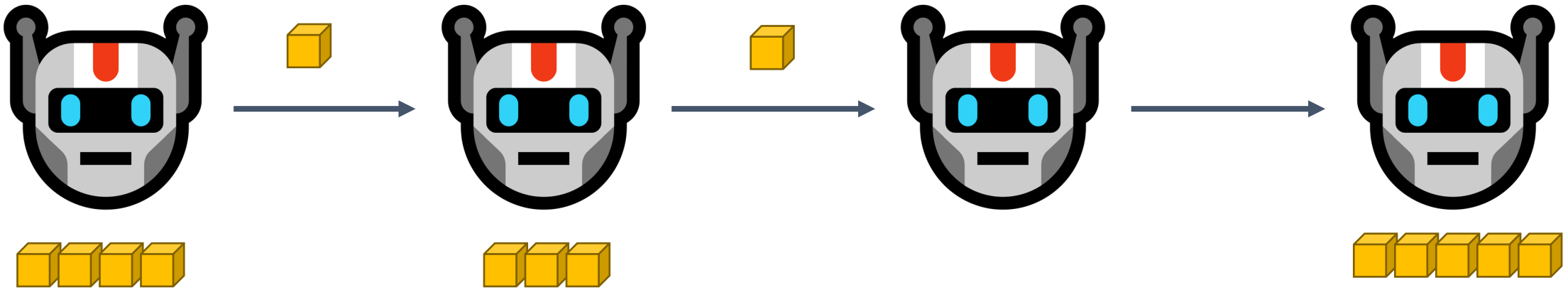
正答数:　26チーム



問題４　荷物渡し１

概要

- N体のロボットが横一列に並んでいる。i体目のロボットはA_i個の荷物を持っている。
- 各ロボットは毎単位時間、荷物を持っているならば右のロボットに荷物を渡す。
- S_j時間後、P_j体目のロボットにB_j個の荷物を渡すことをM回行う。
- T時間後の各ロボットが持っている荷物の数を出力せよ。



問題４　荷物渡し１

制約

- $2 \leq N \leq 100$
- $1 \leq M \leq 100$
- $1 \leq T \leq 100$
- $0 \leq A_j \leq 100$
- $1 \leq B_j \leq 100$

制限時間
5 秒

制約（問題９）

- $2 \leq N \leq 200000$
- $1 \leq M \leq 200000$
- $1 \leq T \leq 1000000000$
- $0 \leq A_j \leq 1000000000$
- $1 \leq B_j \leq 1000000000$

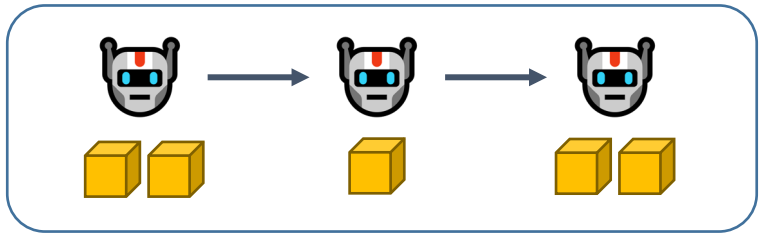
制限時間
2 秒

問題４　荷物渡し１

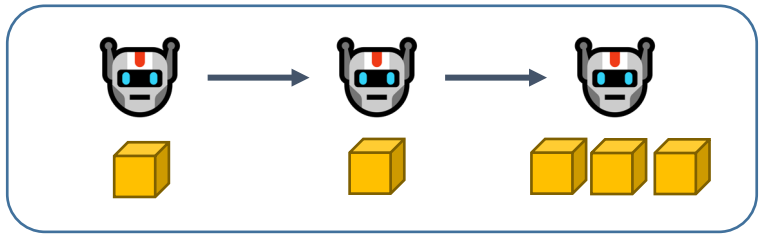
入出力例

入力例１	出力例１
3　1　6 2　1　2 5　1　3	2 1 5

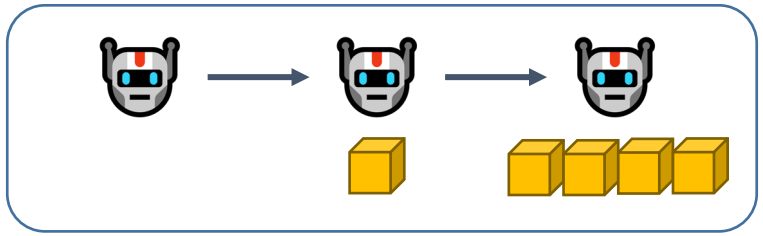
$T = 0$



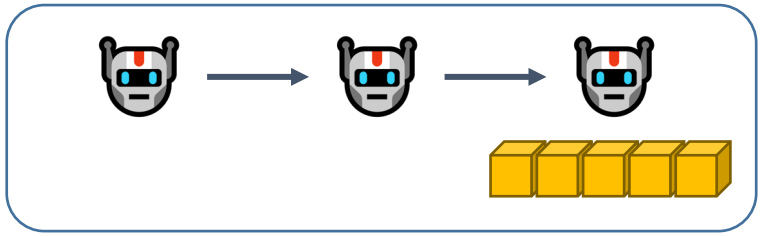
$T = 1$



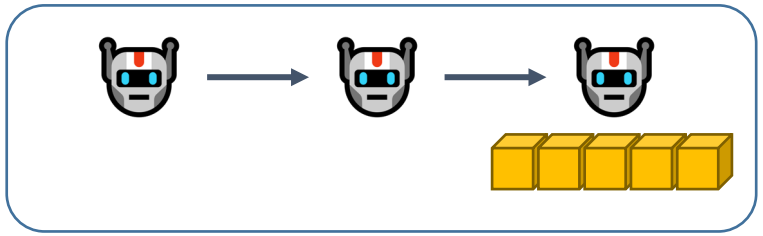
$T = 2$



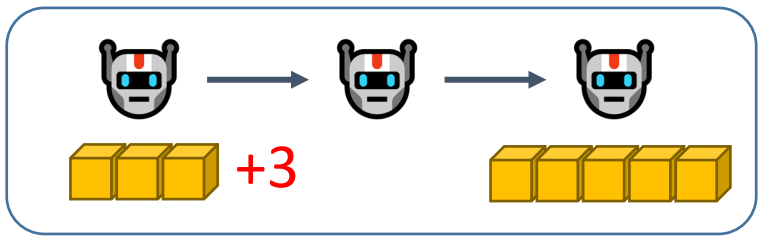
$T = 3$



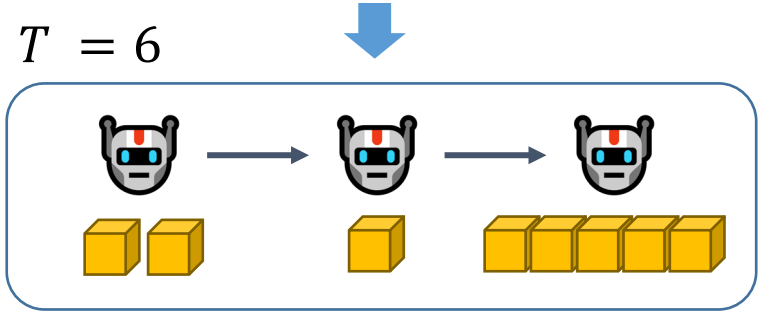
$T = 4$



$T = 5$



$T = 6$



問題４　荷物渡し１

解法

- ロボットの荷物の数を愚直にシミュレーション
- 時間計算量$O\big(T(N\ +\ M)\big)$
- 空間計算量$O(N\ +\ M)$

→　**正解！**

```cpp
#include <iostream>
#include <vector>

int main() {
    int N, M, T;
    std::cin >> N >> M >> T;
    std::vector<int> A(N);
    for (int i = 0 ; i < N ; i++) {
        std::cin >> A[i];
    }
    std::vector<int> S(M), P(M), B(M);
    for (int i = 0 ; i < M ; i++) {
        std::cin >> S[i] >> P[i] >> B[i];
        P[i]--;
    }
    for (int time = 1 ; time <= T ; time++) {
        std::vector<int> next_balls = A;
        for (int i = 0 ; i + 1 < N ; i++) {
            if (A[i] > 0) {
                next_balls[i]--;
                next_balls[i + 1]++;
            }
        }
        for (int i = 0 ; i < M ; i++) {
            if (S[i] == time) {
                next_balls[P[i]] += B[i];
            }
        }
        A = std::move(next_balls);
    }
    for (int i = 0 ; i < N ; i++) {
        std::cout << A[i] << '\n';
    }
}
```

問題４　荷物渡し１

解法

- 追加されるタイミングを前計算すると
- 時間計算量$O(TN+M)$
- 空間計算量$O(N\ +\ M+T)$

→　**正解！**

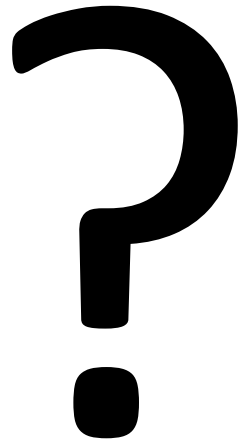
```cpp
#include <iostream>
#include <vector>
#include <utility>

int main() {
    int N, M, T;
    std::cin >> N >> M >> T;
    std::vector<int> A(N);
    for (int i = 0 ; i < N ; i++) std::cin >> A[i];
    std::vector<std::vector<std::pair<int, int>>> add(T + 1);
    for (int i = 0 ; i < M ; i++) {
        int S, P, B;
        std::cin >> S >> P >> B;
        add[S].push_back(std::make_pair(P - 1, B));
    }
    for (int time = 1 ; time <= T ; time++) {
        std::vector<int> next_balls = A;
        for (int i = 0 ; i + 1 < N ; i++) {
            if (A[i] > 0) {
                next_balls[i]--;
                next_balls[i + 1]++;
            }
        }
        for (auto [P, B] : add[time]) {
            next_balls[P] += B;
        }
        A = std::move(next_balls);
    }
    for (int i = 0 ; i < N ; i++) {
        std::cout << A[i] << '\n';
    }
}
```

問題４　荷物渡し１

解法

- 問題９を先に解いて、同じコードを提出する

→　**正解！**



問題４　荷物渡し１

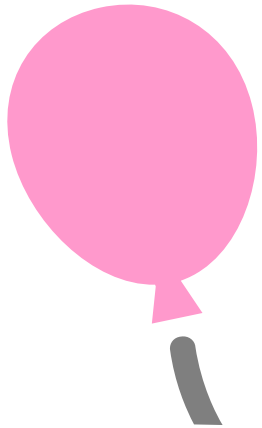
実装上の注意

- すべての荷物の受け渡しは「同時」に行われる
- 前のロボットからもらった荷物を同じ時間に渡さないようにする必要がある
  - ✓ 別の配列を用意する
  - ✓ 後ろからループを回す

```cpp
#include <iostream>
#include <vector>
#include <utility>

int main() {
    int N, M, T;
    std::cin >> N >> M >> T;
    std::vector<int> A(N);
    for (int i = 0 ; i < N ; i++) std::cin >> A[i];
    std::vector<std::vector<std::pair<int, int>>> add(T + 1);
    for (int i = 0 ; i < M ; i++) {
        int S, P, B;
        std::cin >> S >> P >> B;
        add[S].push_back(std::make_pair(P - 1, B));
    }
    // 誤ったシミュレーション
    for (int time = 1 ; time <= T ; time++) {
        for (int i = 0 ; i + 1 < N ; i++) {
            if (A[i] > 0) {
                A[i]--;
                A[i + 1]++;
            }
        }
        for (auto [P, B] : add[time]) {
            A[P] += B;
        }
    }
    for (int i = 0 ; i < N ; i++) {
        std::cout << A[i] << '\n';
    }
}
```

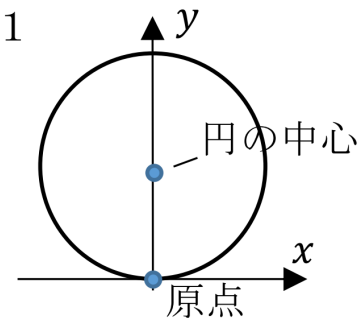
問題５　光線の反射（８点）

正答数:　　28チーム



問題５　光線の反射

図１



概要

- 中心が原点から１メートル上にあり、半径が１メートルの円がある（図１）

1. x軸を基準としてp/q度の角度で光線Lを原点から射出する（図２）。
2. 光線Lが原点に到達すれば終了する。そうでなければ次の3と4をLが原点に到達するまで繰り返す。
3. 光線Lの始点を除きLと円が交差する点を見つけ、それを点Rとする（図３）。
4. 点Rから円の中心に向かうまっすぐな線をTとする（図４）。このとき、光線LとTのなす角度と、点Rから新しく射出する光線とTの角度が同じになるように反射させた光線を、新しく L とする（図５）。

図２

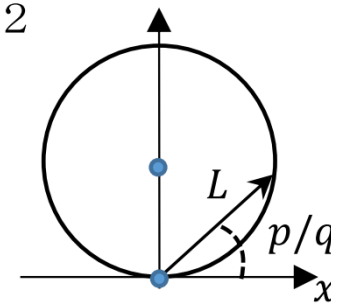


図３

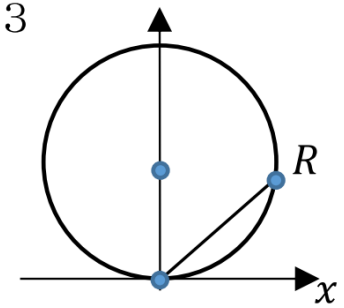


図４

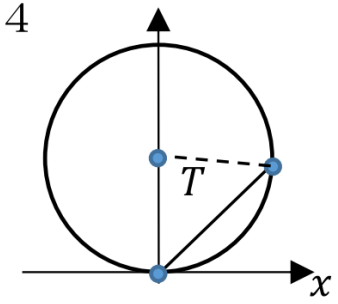
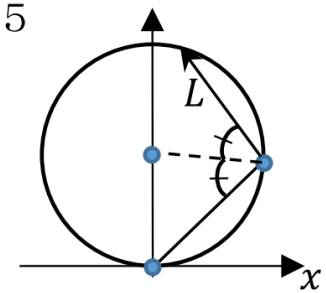


図５

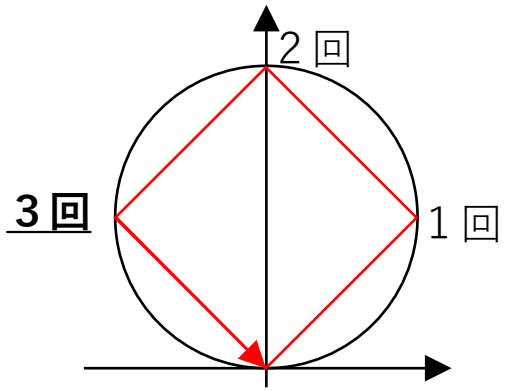


- この手順が終了したとき、光線は何回反射するか？
- 整数pと整数qは$1 \leq p, q \leq 10^{16}$，$0 < p_i/q_i \leq 90$をみたす。
- pとqの最大公約数は１。
- そのようなpとqの組がN個$(1 \leq N \leq 10^5)$与えられる。

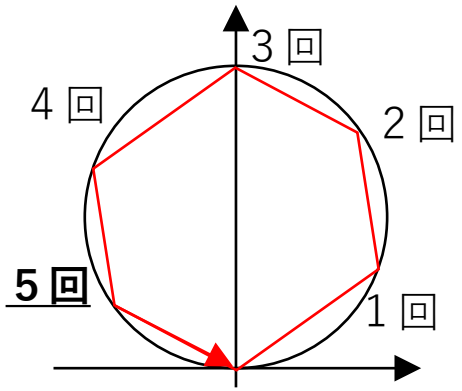
問題５　光線の反射

入出力例

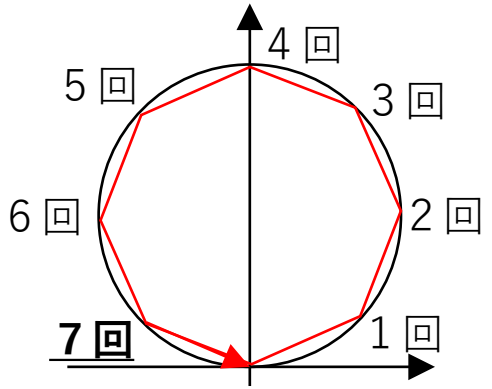
- $p = 45, q = 1 \rightarrow p/q = 45$度



- $p = 30, q = 1 \rightarrow p/q = 30$度



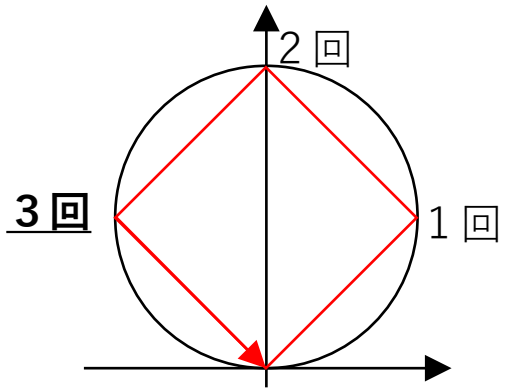
- $p = 45, q = 2 \rightarrow p/q = 22.5$度



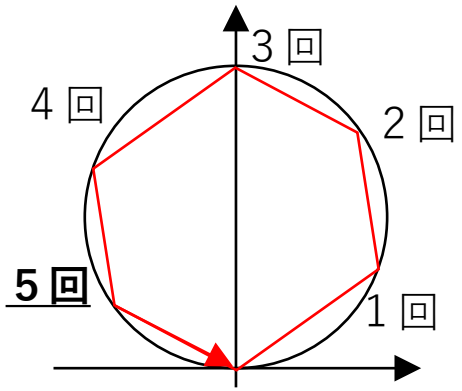
問題５　光線の反射

入出力例

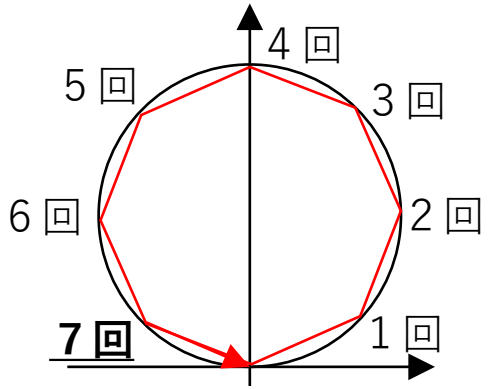
- $p = 45, q = 1 \rightarrow p/q = 45$度



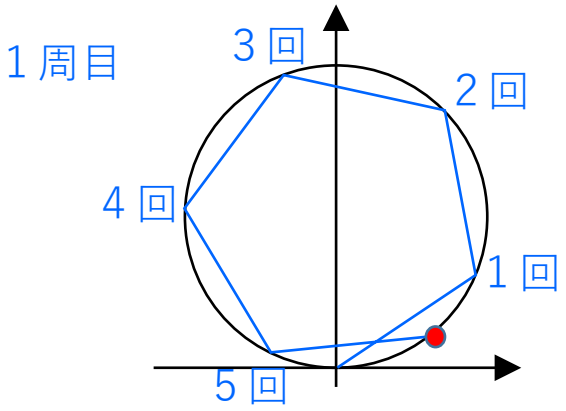
- $p = 30, q = 1 \rightarrow p/q = 30$度

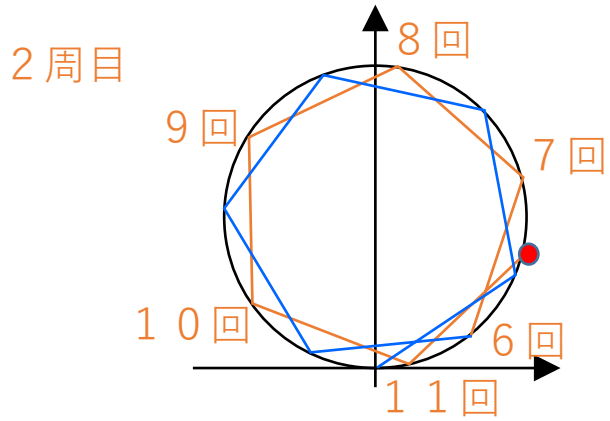


- $p = 45, q = 2 \rightarrow p/q = 22.5$度

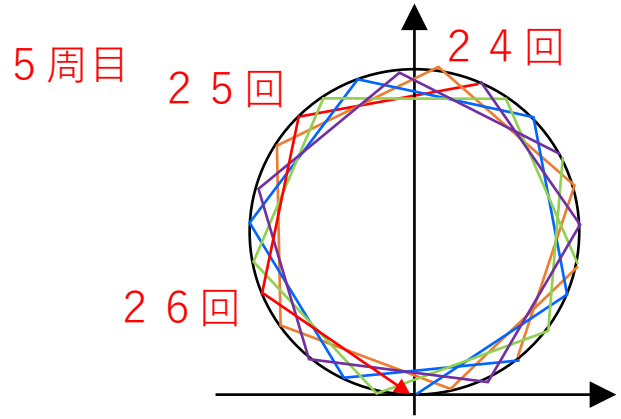


- $p = 100, q = 3 \rightarrow p/q = 33.33\ldots$度　（１周で終わらない場合もあることに注意する）





……



問題５　光線の反射

考察

- 光線の傾き角度の累積が180の倍数になれば、光線は原点に戻る。
- 原点に戻るまでに反射する回数+1をk(正の整数)とする。
- 原点に戻るまでに光線が何周するかをm (正の整数)とする。
- $k\frac{p}{q} = 180m$を満たす最小のm を見つけることができれば、 kも求まる。

式を変形すると　$k = \frac{180q}{p}m$

pとqは共通因数を持たないため、 kが整数になる(pを分母から消す) には$\frac{180}{p}m$が整数である必要がある。

pを分母から消すようなmの因数として、180 と pの共通因数は必要ない。

例えば
$p = 462 = 2 \times 3 \times 7 \times 11$なら、　$\frac{180}{p}m = \frac{2 \times 2 \times 3 \times 3 \times 5}{2 \times 3 \times 7 \times 11}m = \frac{2 \times 3 \times 5}{7 \times 11}m$　なので、 $m = 7 \times 11$

一般に、$m = \frac{p}{\gcd(180,p)}$

問題５　光線の反射

解法

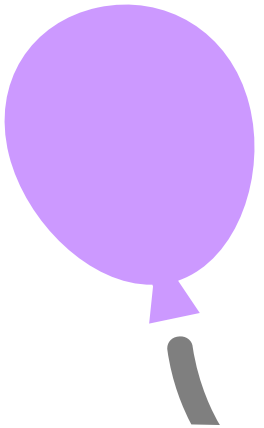
- $k = \frac{180q}{p} m = \frac{180q}{p} \times \frac{p}{\gcd(180,p)} = \frac{180q}{\gcd(180,p)}$
- kは 原点に戻るまでに反射する回数+1なので、$k-1$が答え。

解答例（C++）

```
int main(){
  int n;
  cin>>n;
  while(n--){
    long long p,q;
    cin>>p>>q;
    cout<<q*180/gcd(p,180)-1<<'¥n';
  }
}
```

問題６ 荷物の配置パターン（１０点）

正答数:　　18チーム



問題６　荷物の配置パターン

概要

- 集積所にN個の荷物がある。i番目の荷物の重さはB_i であり、倉庫A_iに移動させることができる。
- 各荷物は集積所に置いたままにするか、倉庫に移動する→　荷物の配置パターンは2^N通り
- 各配置パターンについて、以下のようにして求まる「最大重量」を計算する。
  - ➢それぞれの倉庫ごとに置かれた荷物の重さの合計を計算する。
  - ➢合計の中での最大値を、その配置パターンの最大重量と呼ぶ。
- このとき、最大重量がkとなるような配置パターンの総数を、$f(k)$とする
- $k = 0,1,\cdots,M$の場合について$f(k)$を求めよ（998244353で割った余り）

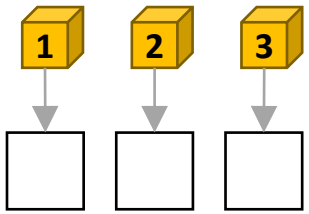
制約

- $N\ (1 \leq N \leq 300)$、$M\ (1 \leq M \leq 100{,}000)$、$A_i\ \ (1 \leq A_i \leq N)$、$B_i\ \ (1 \leq B_i \leq 300)$

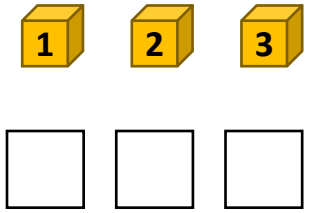
問題６　荷物の配置パターン

入出力例１

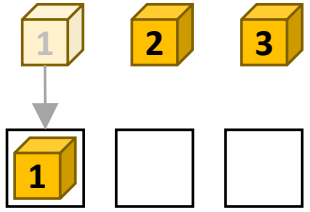
入力例１	出力例１
3　3 1　1 2　2 3　3	1 1 2 4



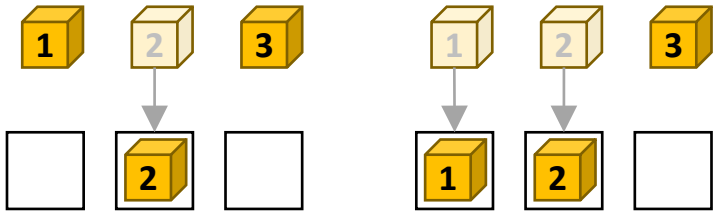
$f(0) = 1$



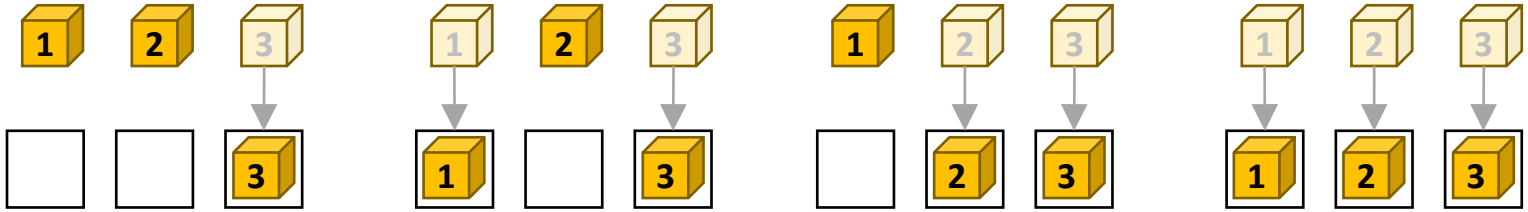
$f(1) = 1$



$f(2) = 2$



$f(3) = 4$

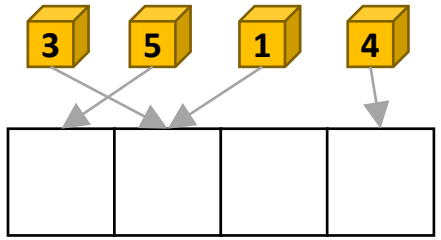


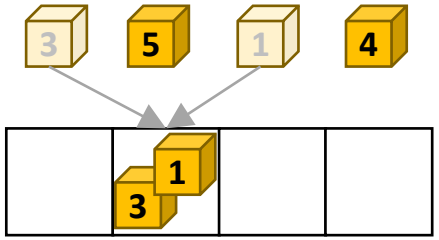
問題６　荷物の配置パターン

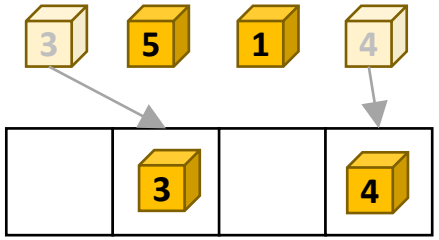
入出力例2

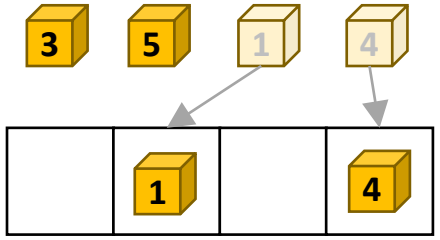
入力例2	出力例2
4 5 2 3 1 5 2 1 4 4	1 1 0 1 5 8

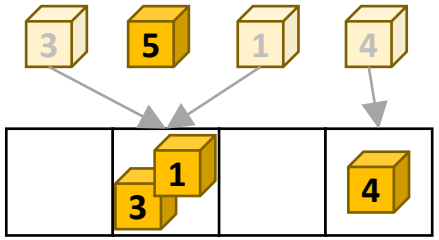
$f(4) = 5$

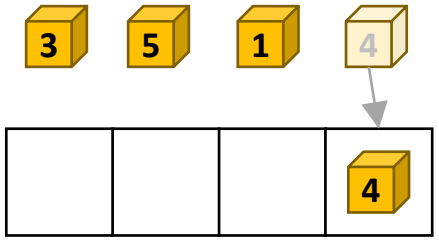








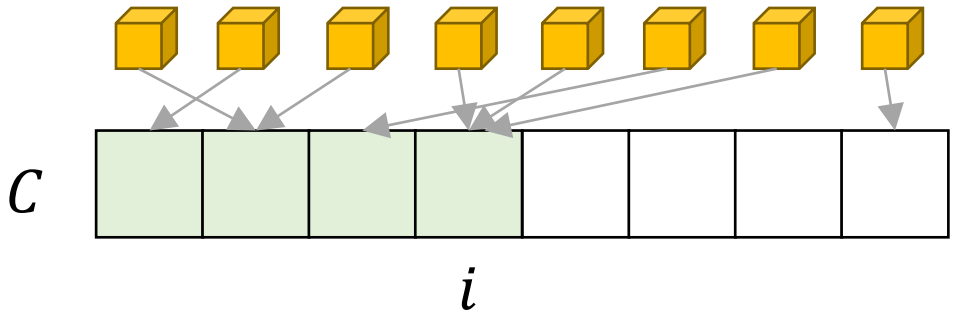




問題６　荷物の配置パターン

解法（動的計画法）

- 倉庫の荷物の重さの配列をCとする
- $f(k)$
  ＝「最大値がkとなる選び方の総数」
  ＝「Cの最大値がk以下となる選び方の総数」－「Cの最大値がk未満となる選び方の総数」



$\prod_{i=1}^{N}$「$A_j = i$を満たす操作(A_j, B_j)のみで、C_iをk以下にする選び方の総数」…①

$\prod_{i=1}^{N}$「$A_j = i$を満たす操作(A_j, B_j)のみで、C_iをk未満にする選び方の総数」…②

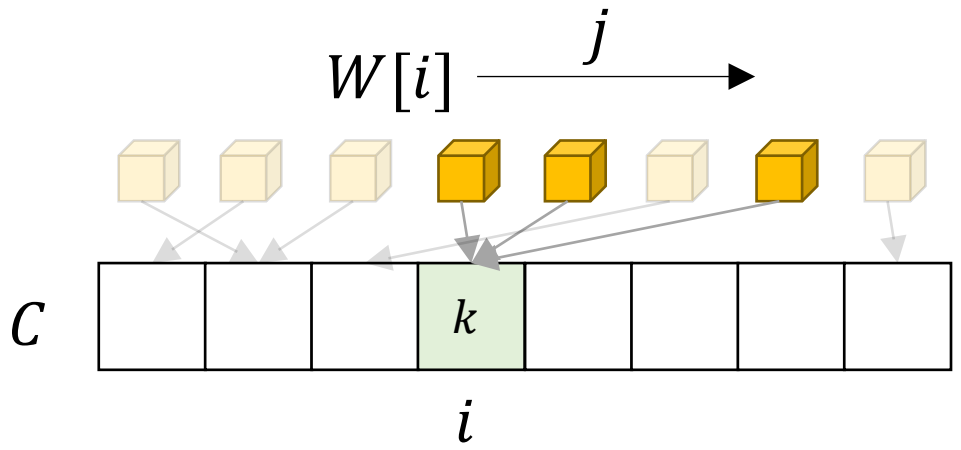
①－②で求める

問題６　荷物の配置パターン

解法（動的計画法）

$\mathrm{dp}[i][j][k] :=$ 「$(A_x = i, 1 \leq x \leq j)$を満たす操作$(A_x, B_x)$のみで、$C_i$を$k$にする選び方の総数」

```
...
// ※ iの次元を省略
for ( int j = 0; j < W[i].size(); j++ ){
  for ( int k = 0; k <= M; k++ ){
    dp[j+1][k] += dp[j][k]; // ※ MOD省略
    dp[j+1][k + W[i][j]] += dp[j][k];
  }
}
```



問題６　荷物の配置パターン

解法（動的計画法）

$\mathrm{dp}[i][j][k] :=$ 「$(A_x = i, 1 \leq x \leq j)$を満たす操作$(A_x, B_x)$のみで、$C_i$を$k$にする選び方の総数」

$\mathrm{cum}[i][k] := \sum_{j=0}^{k} \mathrm{dp}[i][N][j] \cdots$　　$A_j = i$を満たす操作(A_j, B_j)のみで、C_iをk以下にする選び方の総数

$\mathrm{cum}[i][k-1] \cdots$　　$A_j = i$を満たす操作(A_j, B_j)のみで、C_iをk未満にする選び方の総数

$f(k) = \prod_{i=1}^{N} \mathrm{cum}[i][k] - \prod_{i=1}^{N} \mathrm{cum}[i][k-1]$

問題６　荷物の配置パターン

計算量

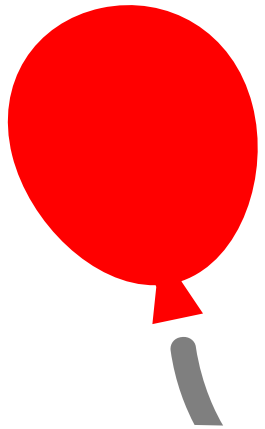
- $\mathrm{dp}[i][j][k]$の構築に$O(N\sum B)$
- M回cumの積を求めるのに$O(NM)$
- 全体で$O\big(N(M+\sum B)\big)$

注意点

- MODの取り忘れ
- 無駄にテーブルを作ると、**メモリ制限**

問題７盆栽育成ゲーム（１０点）

正答数:　　10チーム



問題７　盆栽育成ゲーム

概要

- N頂点からなる木が与えられる。各頂点iには整数A_iが書き込まれている。
- 与えられた木に対して、次数1の頂点を一つ選び、選んだ頂点とそれに接続する辺を削除することを、0回以上行うことができる。
- 操作後に残った木の頂点の集合をV、頂点vの次数を$\deg(v)$とするとき、

$$\sum_{v \in V} \deg(v) A_v$$

※問題文ではa_i

をこの木の芸術的価値とする。

- 操作後に残った木の芸術的価値の最大値を求めよ。

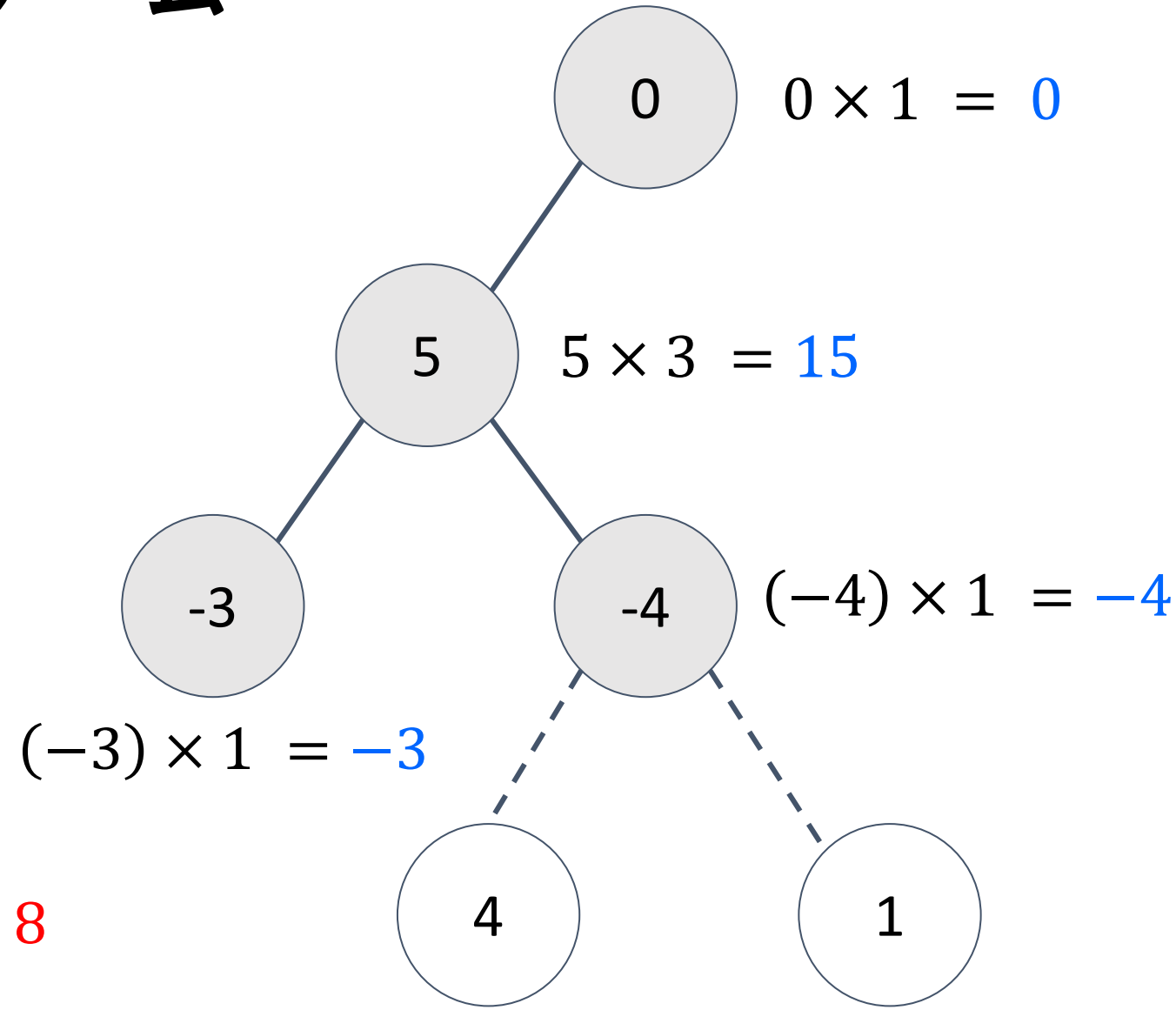
制約

- $2 \leq N \leq 10^5$
- $-2 \times 100000 \leq A_i \leq 2 \times 100000$

問題７　盆栽育成ゲーム

入出力例１

入力例１	出力例１
6 1 2 2 3 2 4 3 5 3 6 0 5 -4 -3 4 1	8



$0 + 15 - 3 - 4 = 8$

問題７　盆栽育成ゲーム

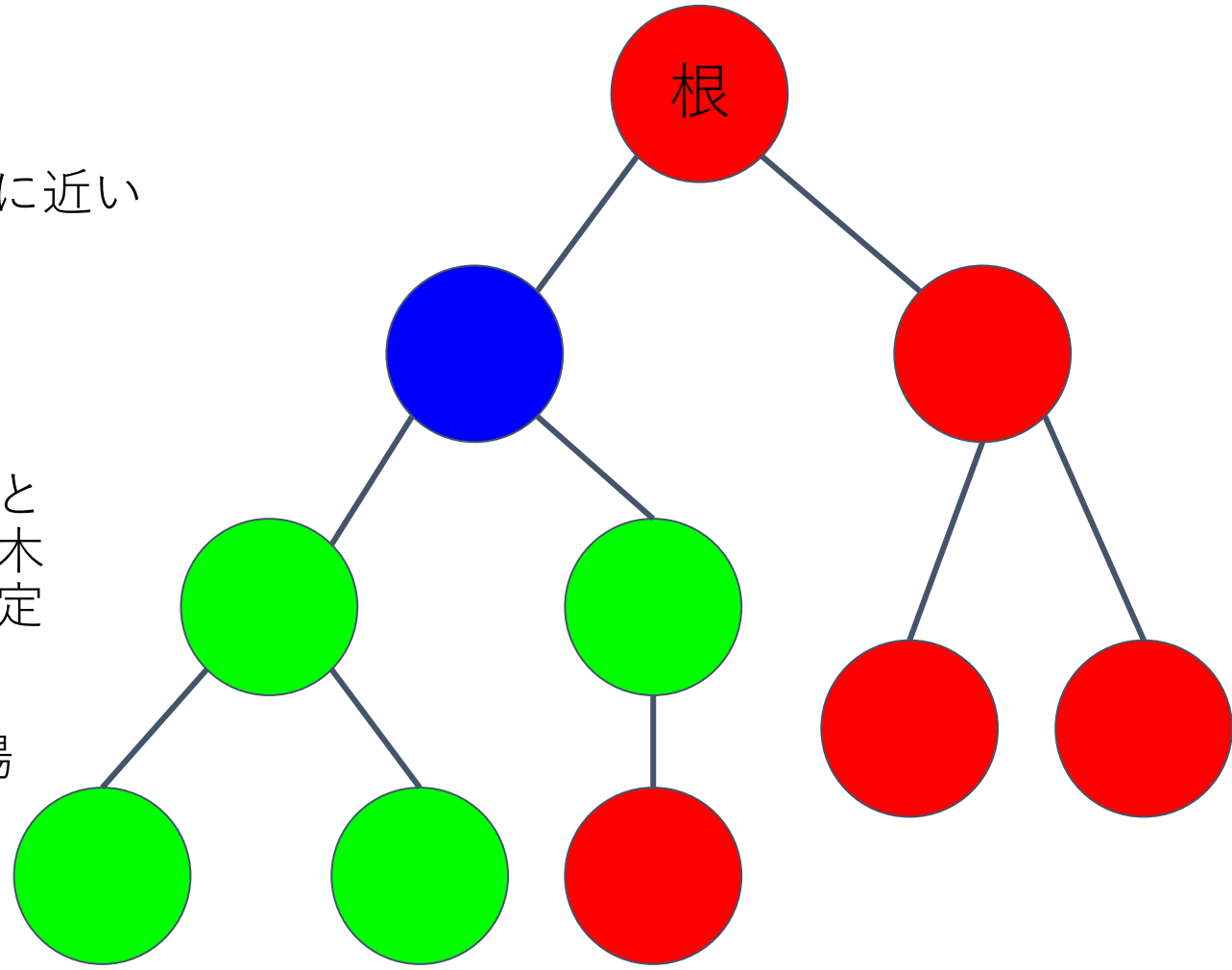
方針

根を固定する。消さない頂点であって、最も根に近いものを決めてこれを全探索　👉 木DP

考察

与えられた木を適当な頂点を根とした根付き木とみなす。どのような作業においても、作業後の木について、「最も根に近い頂点」が**ただ一つ**に定まる。

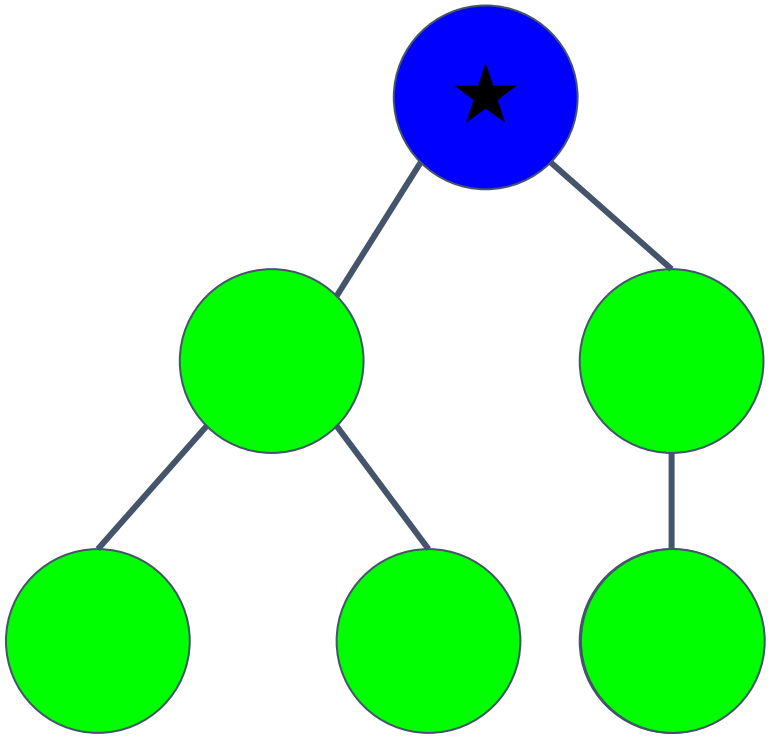
- 右図で、赤く塗った頂点を作業で消した場合は青く塗った頂点が該当



問題７　盆栽育成ゲーム

考察

「残す頂点の中でもっとも根に近い頂点(★)」を決め打ったときの芸術的価値としてありうる最大値を考える。



💡★より祖先の頂点は**必ず消すことが確定している**
-> 考慮しなくて良い

💡★以外の残っている頂点は、**その親が必ず残っている**（緑の頂点）

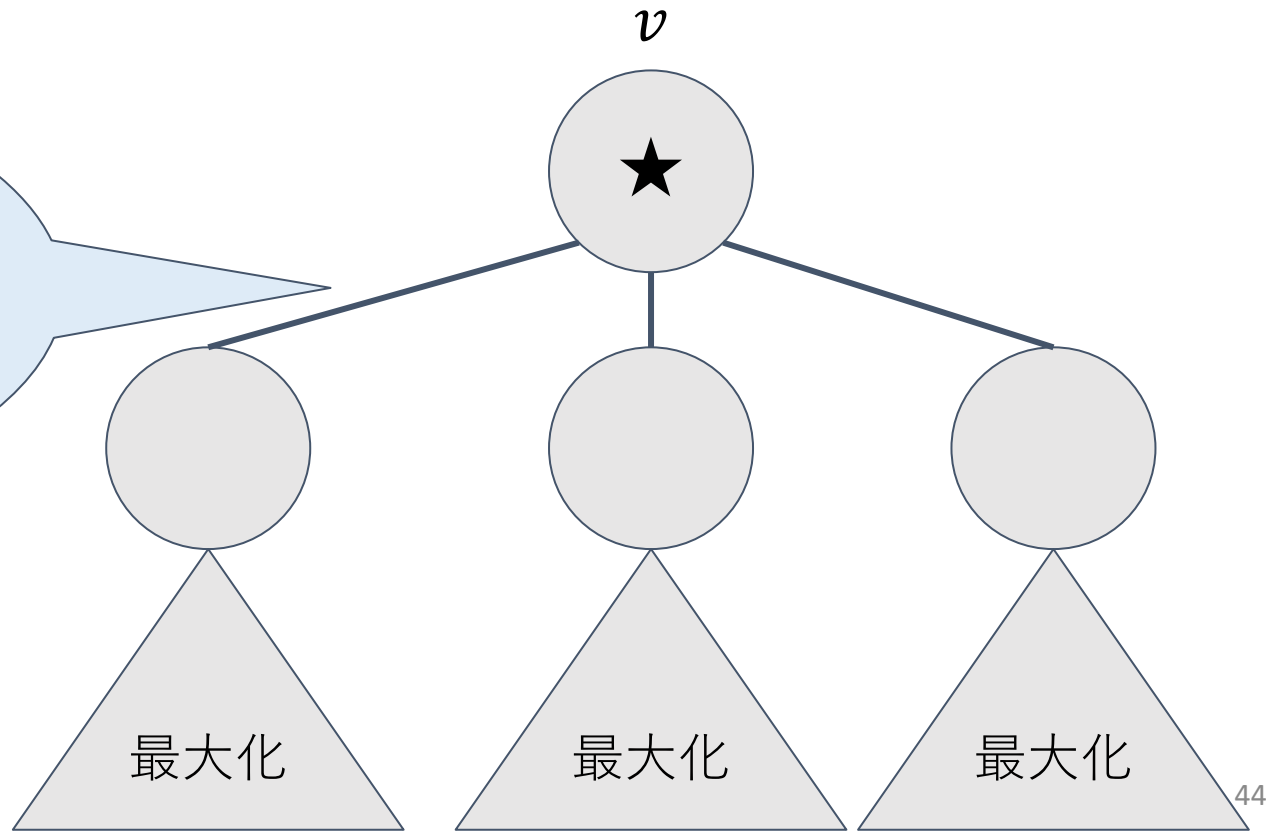
問題７　盆栽育成ゲーム

考察

頂点vを★にしたとする。頂点vからの寄与を考える。

vのある子頂点を残すとき、 vからの寄与が丁度A_v増える。

それぞれの子頂点の部分木からの寄与を最大化して、
寄与が$-A_v$ 以上なら残し、
そうでないなら残さず部分木の頂点を全部消すのが最適

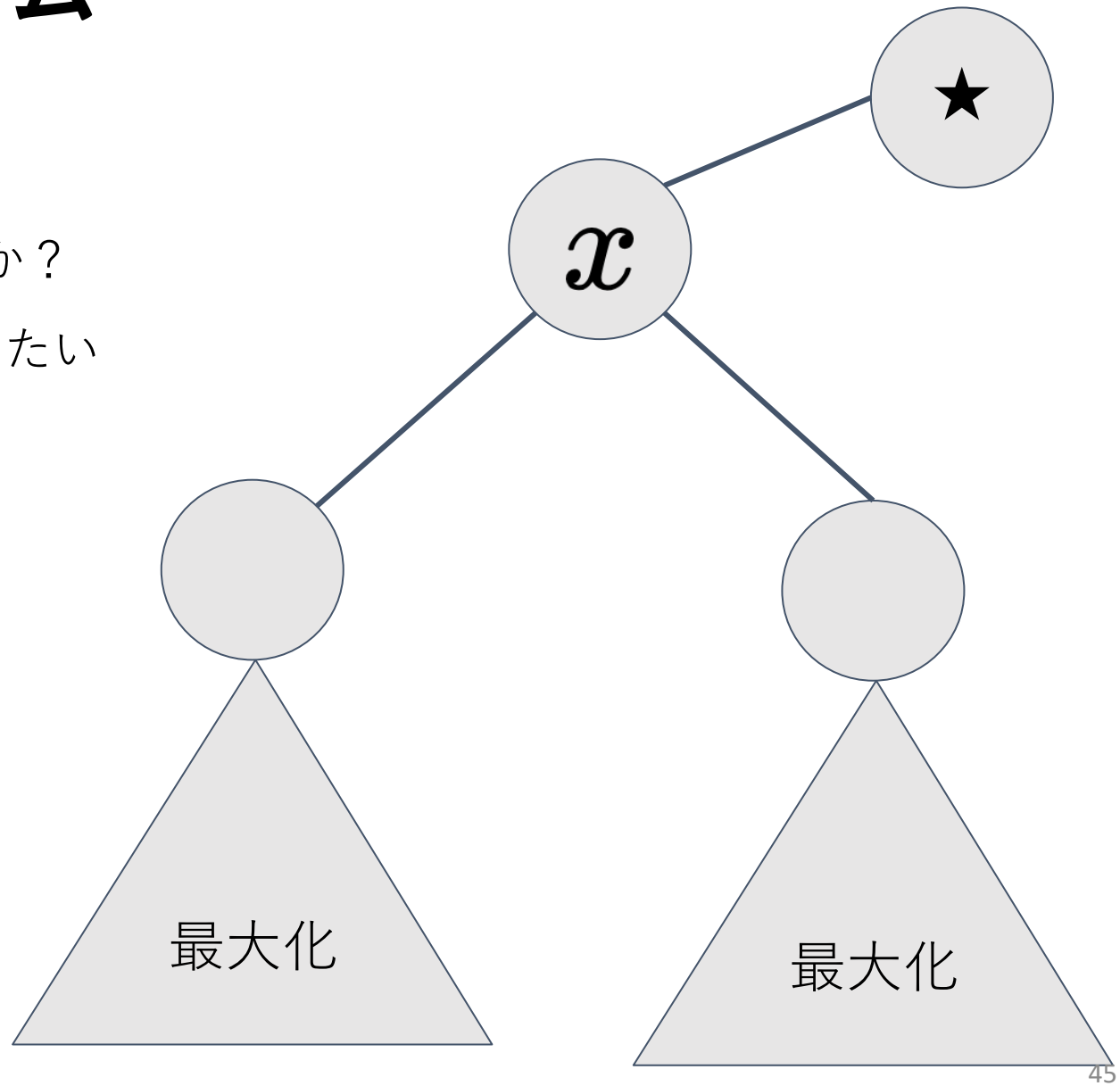


問題７　盆栽育成ゲーム

考察

部分木からの寄与はどうすれば最大化されるか？

頂点xを根とした部分木からの寄与を最大化したい



💡親は必ず残るので、A_xは寄与する

💡xの子に関しては、★と同じ議論

子の部分木からの寄与を最大化して、
その寄与が$-A_v$以上なら残す。
そうでないなら残さない。

問題７　盆栽育成ゲーム

考察

部分木からの寄与を最大化するためには、子頂点を根とする部分木からの寄与を最大化する必要がある。

- 部分木毎に独立に考えて良い
- サイズ(頂点数)が小さい同様の問題に帰着　→　💡**動的計画法**

解法：動的計画法

dp_v = 親頂点は必ず残るという仮定の下、頂点vを根とした部分木からの寄与の最大値

vの子頂点の集合をC_vとして、

$$\mathrm{dp}_v = A_v + \max_{C' \subseteqq C_v} \left(\sum_{x \in C'} (\mathrm{dp}_x + A_v) \right)$$ と計算できる。

問題７　盆栽育成ゲーム

解法：動的計画法

$$\mathrm{dp}_v = A_v + \max_{C' \subseteqq C_v} \left(\sum_{x \in C'} (\mathrm{dp}_x + A_v) \right)$$

親頂点が残るから

$dp_x \geq -A_v$なる頂点のみを全てC'に含めれば最大化される

葉頂点から親へ登っていく順番にdpを計算することができる

問題７　盆栽育成ゲーム

解の計算

★を頂点vに決め打ったときの芸術的価値の最大値は

$$\max_{C' \subseteqq C_v} \left(\sum_{x \in C'} (\mathrm{dp}_x + A_v) \right)$$

dpの計算とほぼ同じ
親頂点は必ず消すので、その分の寄与A_vが無いだけ

これらの値も葉から親へのぼる順番に計算できる

問題７　盆栽育成ゲーム

計算量

dp_vや★を固定したときの芸術的価値の計算には$O(|C_v|)$　かかる

$\sum_v |C_v| = N - 1$なので全体の時間計算量は$O(N)$

注意点

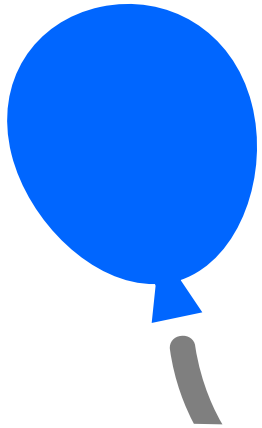
頂点が１つの場合、芸術的価値が0になることに注意　→　解は必ず0以上になる

別解

全方位木DP (考察易・実装難)

問題８　多角形の重なり（１０点）

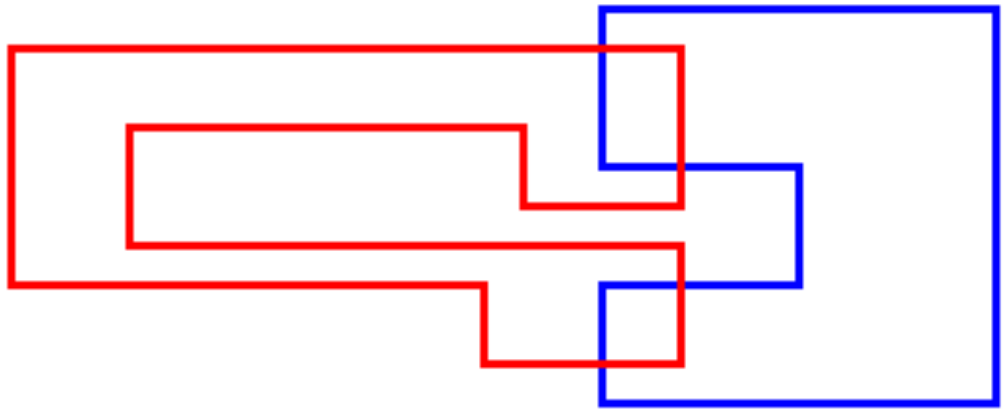
正答数:　　6チーム



問題８　多角形の重なり

概要

- それぞれN, M個の頂点なる２つの多角形が与えられる。両方とも以下の条件を満たす。
  - 全ての頂点が格子点上にある
  - 単純である
  - 全ての角は直角である
  - 全ての辺がx軸またはy軸に平行である
- この２つの多角形の共通部分の面積を求めよ
- $4 \leq N, M \leq 1500$
- 頂点の座標(x, y)の範囲は $-10^9 \leq x_i, y_i \leq 10^9$



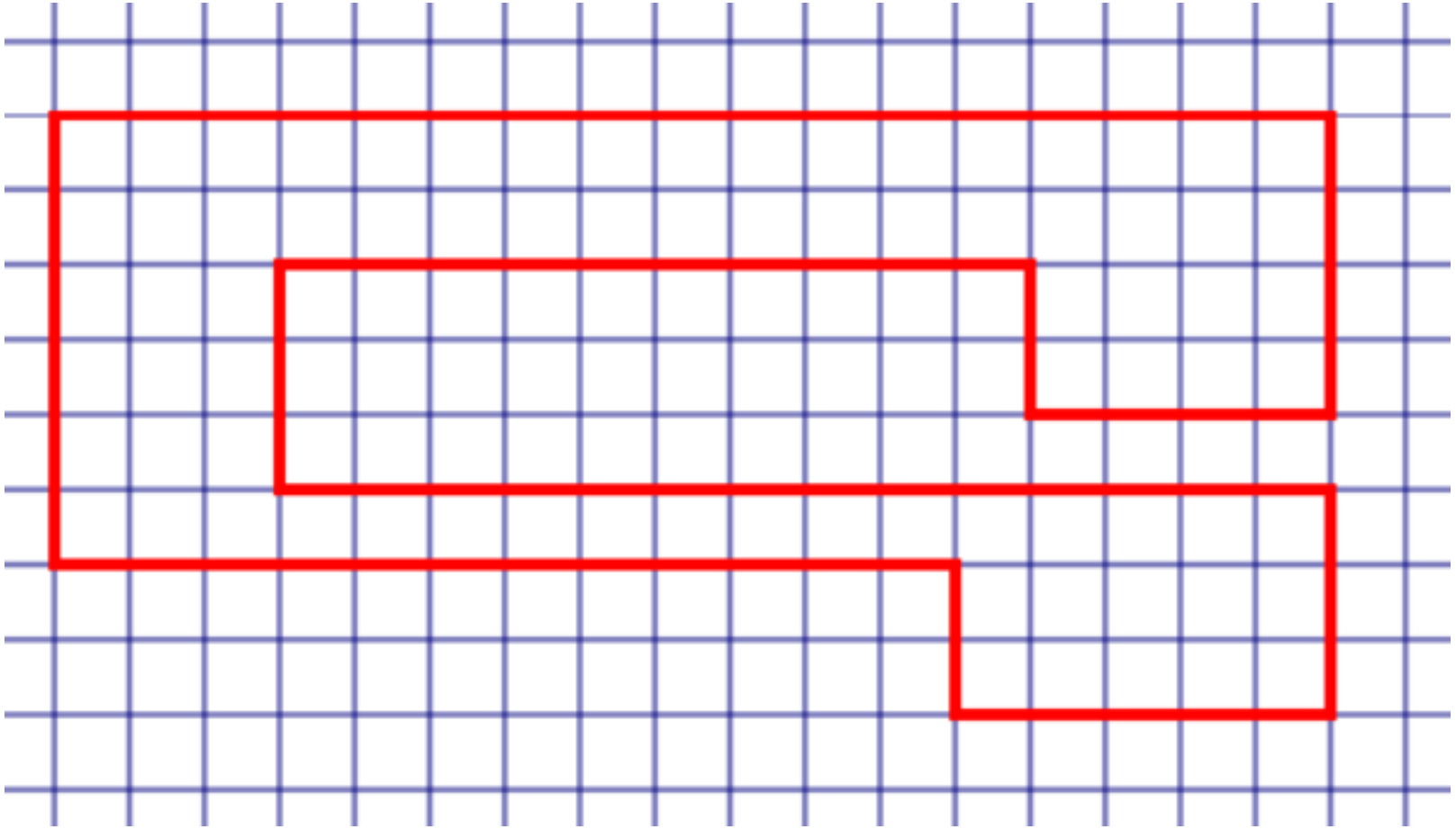
問題８　多角形の重なり

考察

多角形の良い性質を探る

方眼紙上に多角形を描画すると、多角形のすべての辺は方眼紙の線上に乗る

多角形内の領域は方眼紙のマスで分割することができる



問題８　多角形の重なり

考察

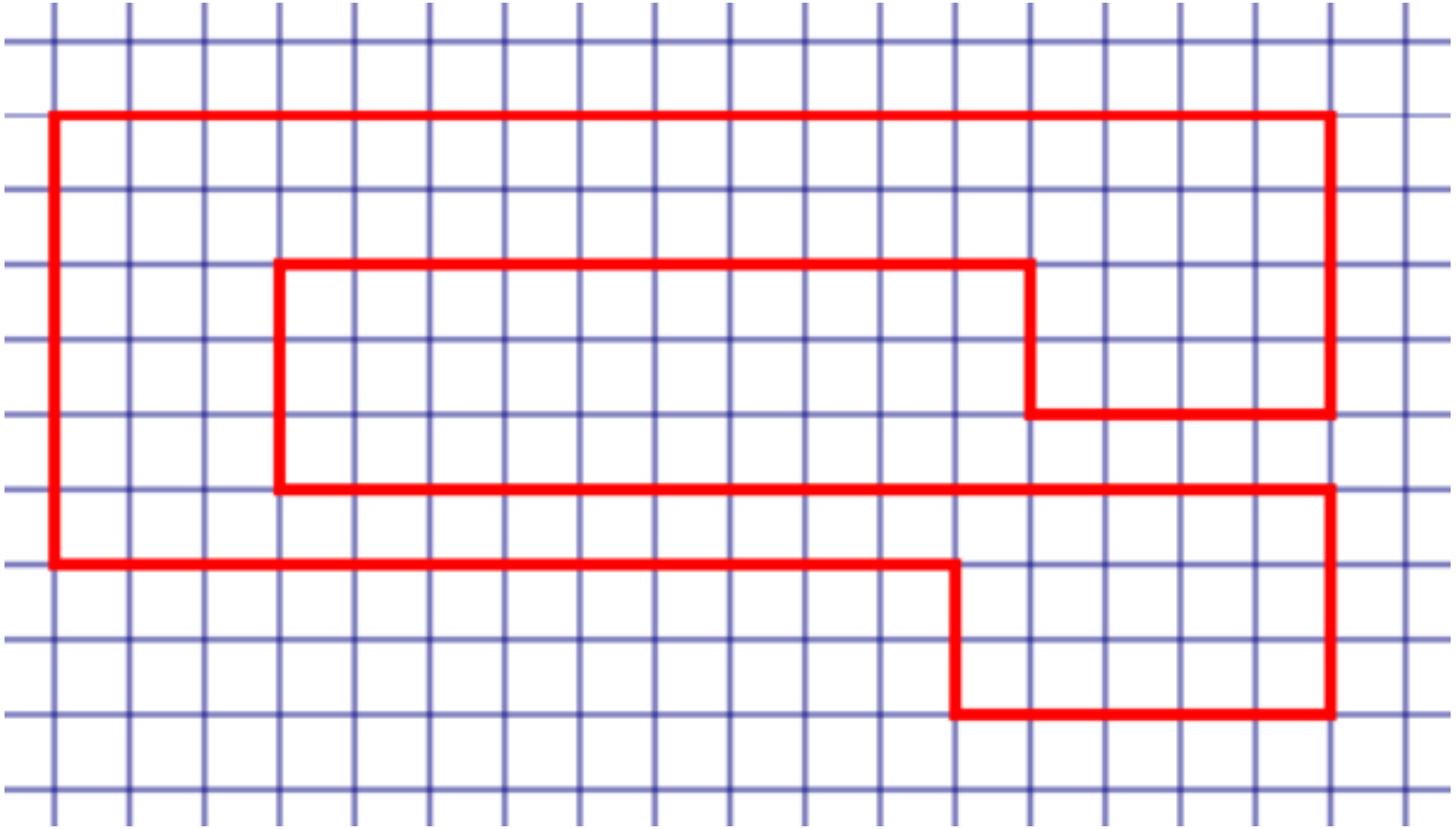
多角形内部にあるマスの数を数える

多角形の面積

=

多角形内部の方眼紙のマスの数

どのようなマスが多角形内部に入るか、特徴を探そう



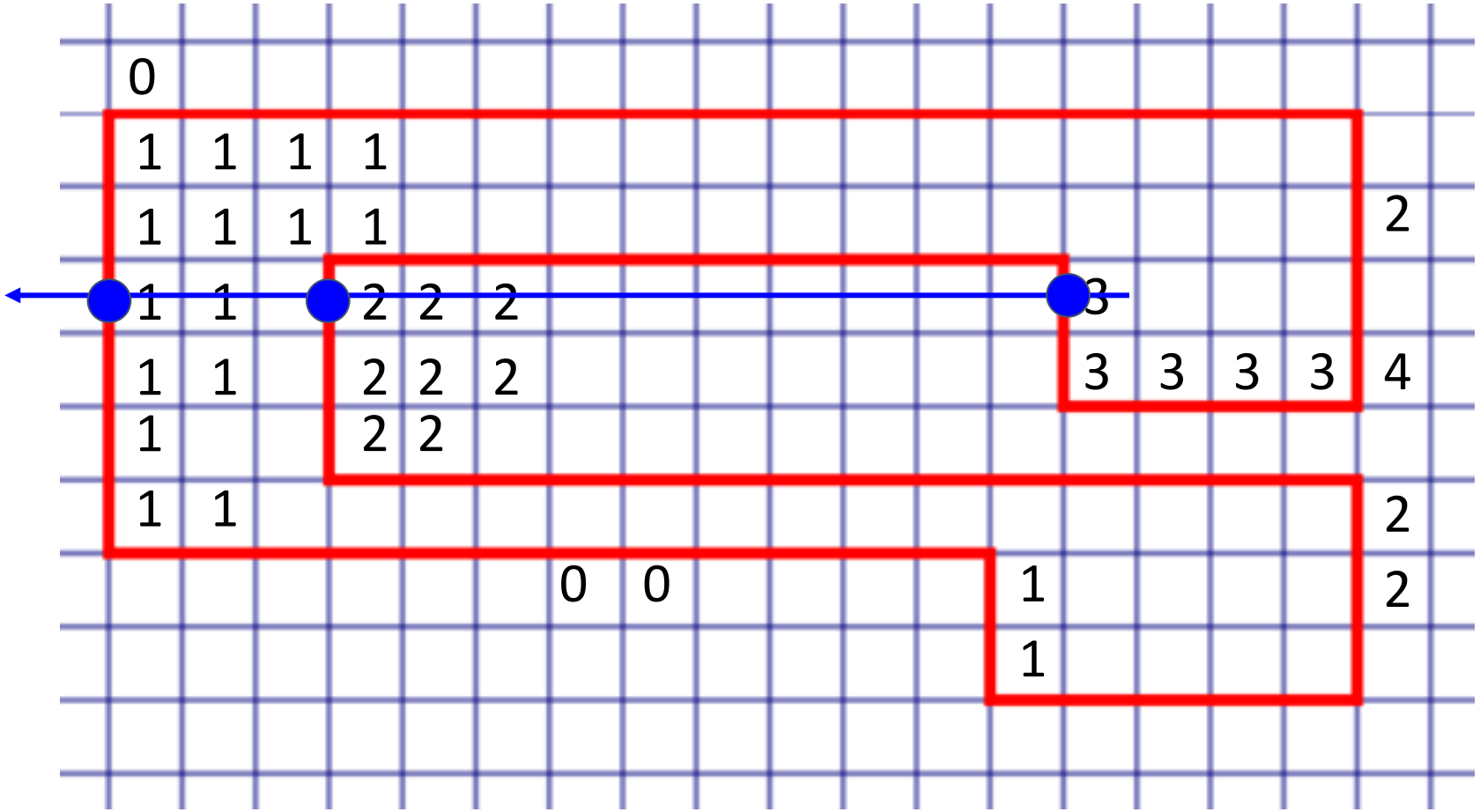
問題８　多角形の重なり

考察

多角形内部にあるマスの特徴付け

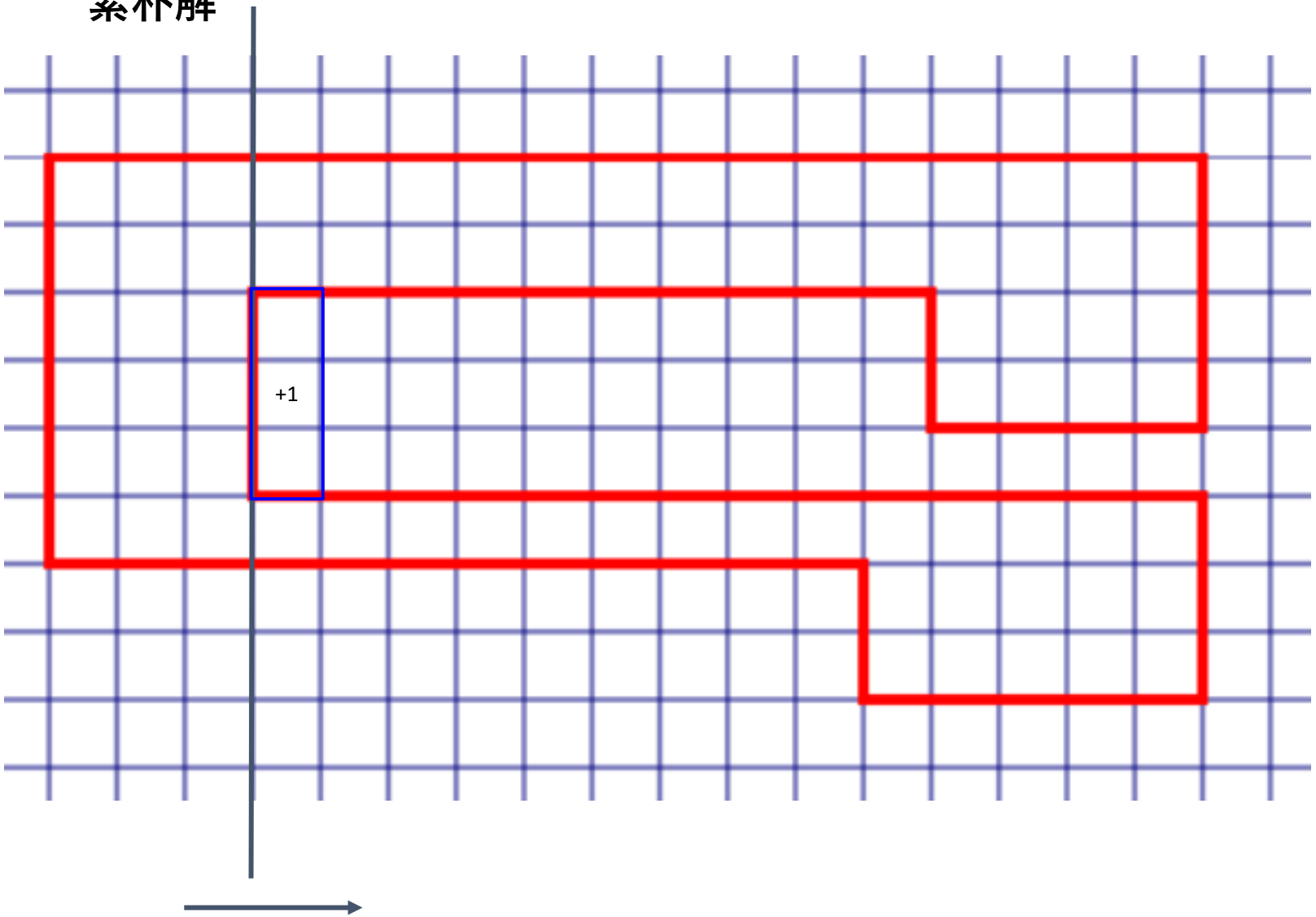
多角形内部にマスがある
→
マスから左側に半直線を伸ばすと、多角形の辺と**奇数回**ぶつかる

これが必要十分条件になっている。



問題８　多角形の重なり

素朴解



x軸負側のマスから多角形の内部に属しているか数えていく

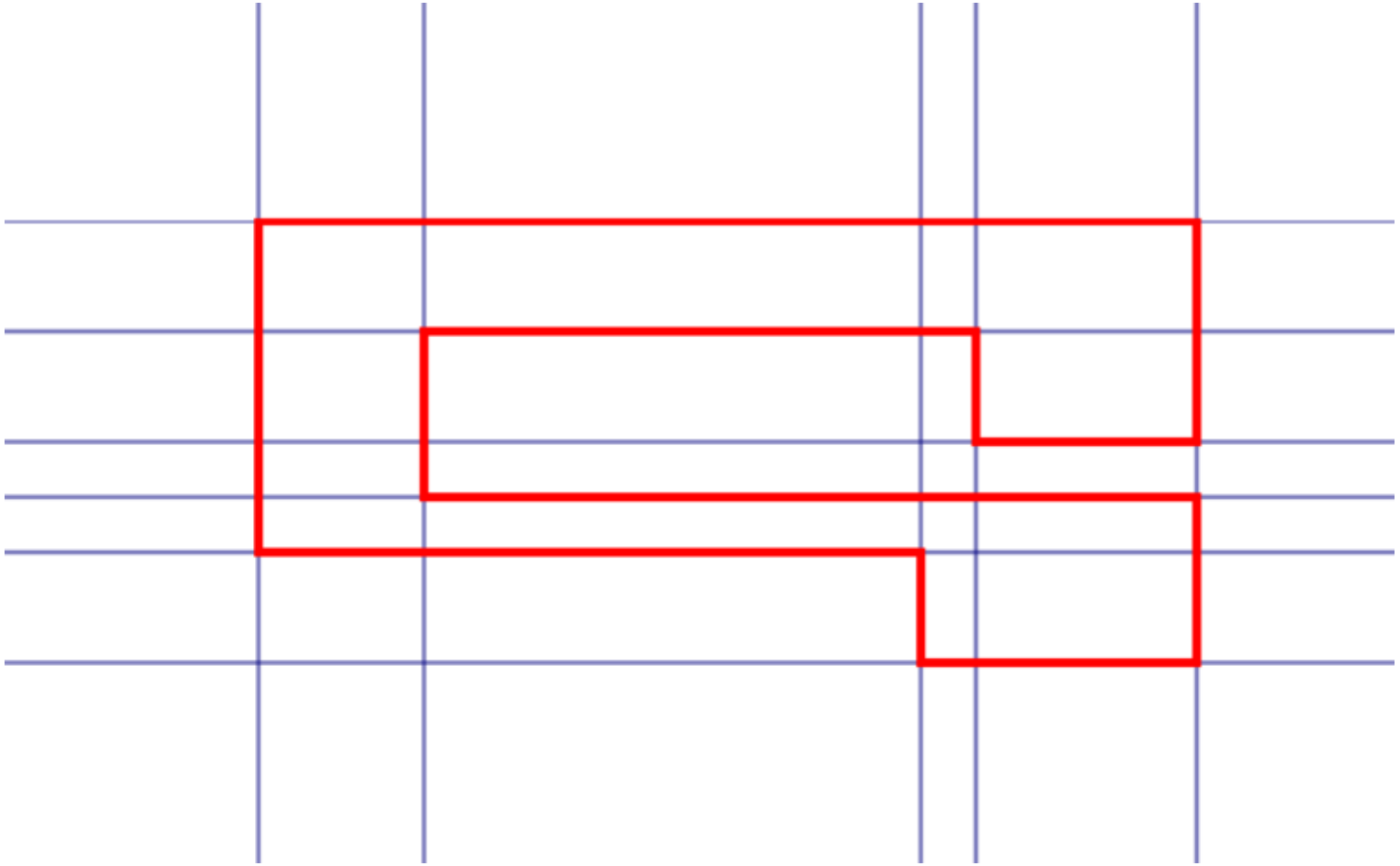
$y = -10^9, \ldots, 10^9$まで、現在まで何回辺にぶつかったかのカウンタを保持する。

多角形のy軸に平行な辺を発見したら、辺と被っている部分のカウンタをインクリメントする

値が奇数になっているカウンタの数だけ解に足し合わせる

問題８　多角形の重なり

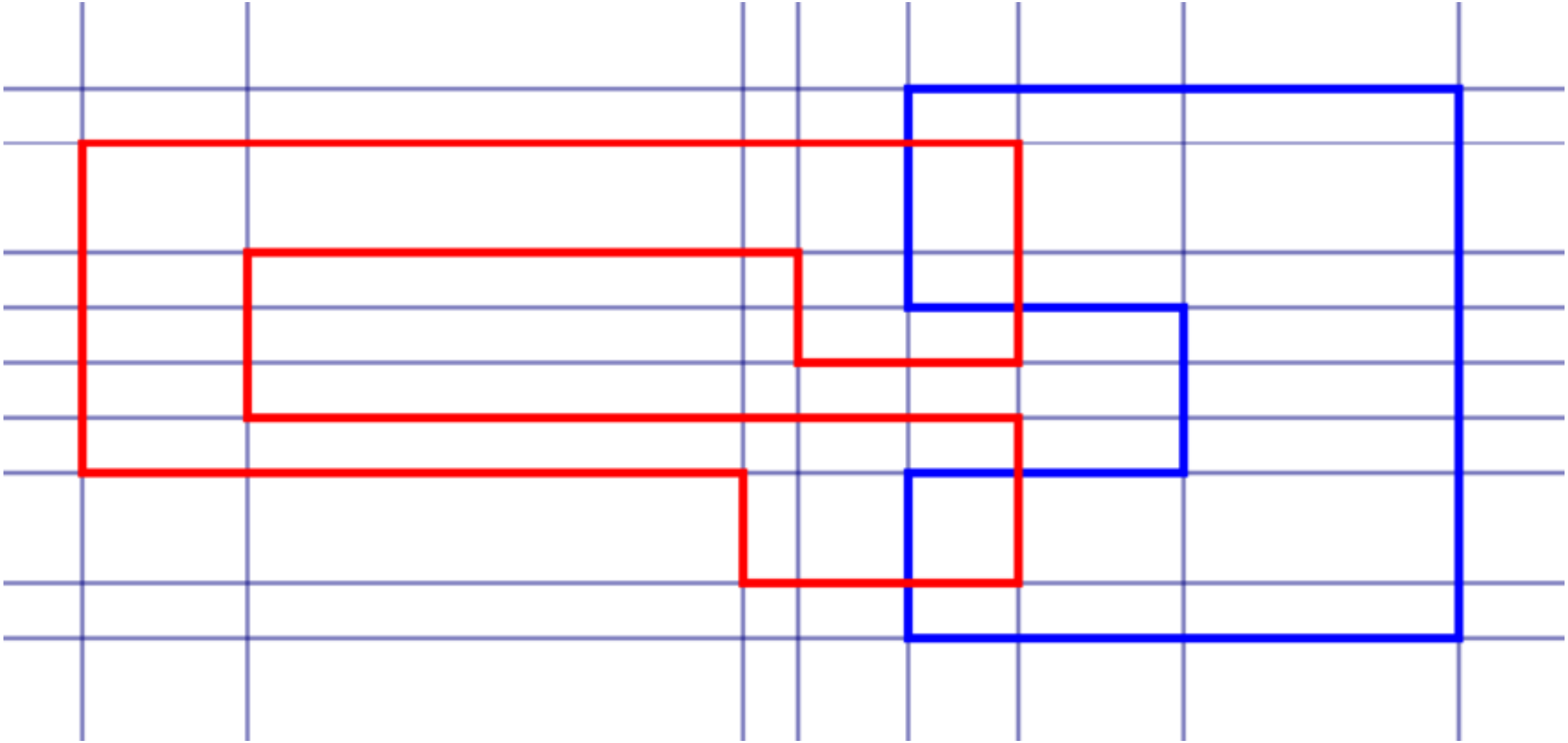
マスの数を減らす



カウンタの値が同じになるような矩形領域を一つにまとめると、考慮すべきマスの数が頂点数の二乗の定数倍に抑えられる

問題８　多角形の重なり

共通部分を求める問題でも同じ



問題８　多角形の重なり

解法のまとめ

- 多角形の頂点として登場するx座標、y座標を列挙して、考慮すべき矩形領域を得る。
- x座標がより負である領域から奇数回辺とぶつかる領域を数える。
- $O((N + M)^2)$で解ける

キーワード

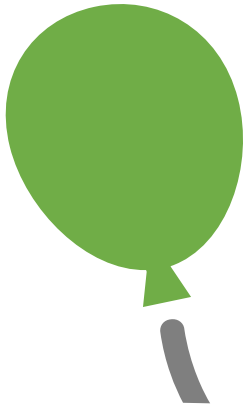
- 座標圧縮
- 平面走査
- 累積和

別解

- 遅延評価付きセグメント木を用いて$O((N + M)\log(N + M))$でも解ける
  - 本問題では過剰

問題９　荷物渡し２　　（１２点）

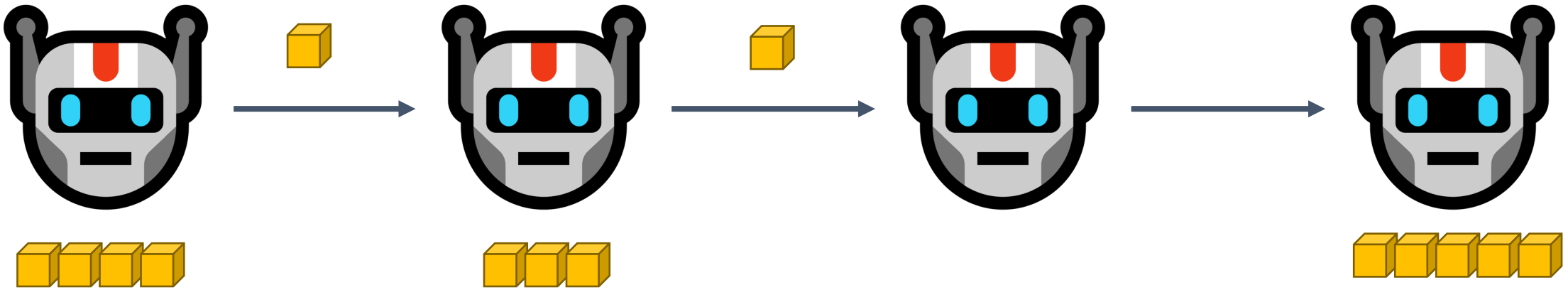
正答数:　　１チーム



問題９　荷物渡し２

概要

- N体のロボットが横一列に並んでいる。i体目のロボットはA_i個の荷物を持っている
- 各ロボットは毎単位時間、荷物を持っているならば右のロボットに荷物を渡す
- S_j時間後、P_j体目のロボットにB_j個の荷物を渡すことをM回行う
- T時間後の各ロボットが持っている荷物の数を出力せよ



問題９　荷物渡し２

制約

- $2 \leq N \leq 100$
- $1 \leq M \leq 100$
- $1 \leq T \leq 100$
- $0 \leq A_j \leq 100$
- $1 \leq B_j \leq 100$

制限時間
5 秒

制約（問題９）

- $2 \leq N \leq 200000$
- $1 \leq M \leq 200000$
- $1 \leq T \leq 1000000000$
- $0 \leq A_j \leq 1000000000$
- $1 \leq B_j \leq 1000000000$

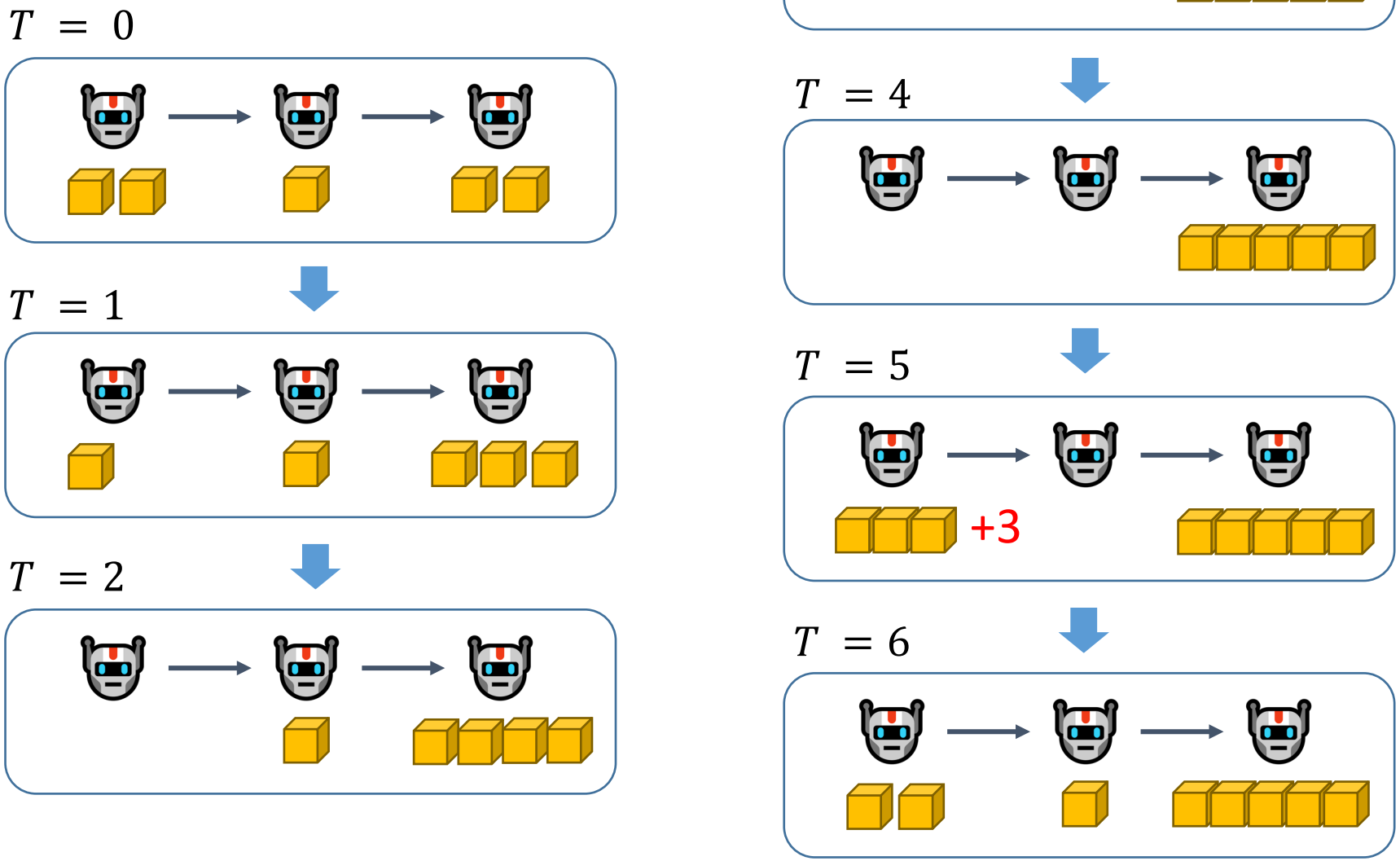
制限時間
2 秒

問題4で説明したシミュレーション解法は時間制限

問題９　荷物渡し２

入出力例

入力例１	出力例１
3 1 6 2 1 2 5 1 3	2 1 5



問題９　荷物渡し２

問題の定式化

i体目のロボットがT時間後に持っている荷物の数

＝

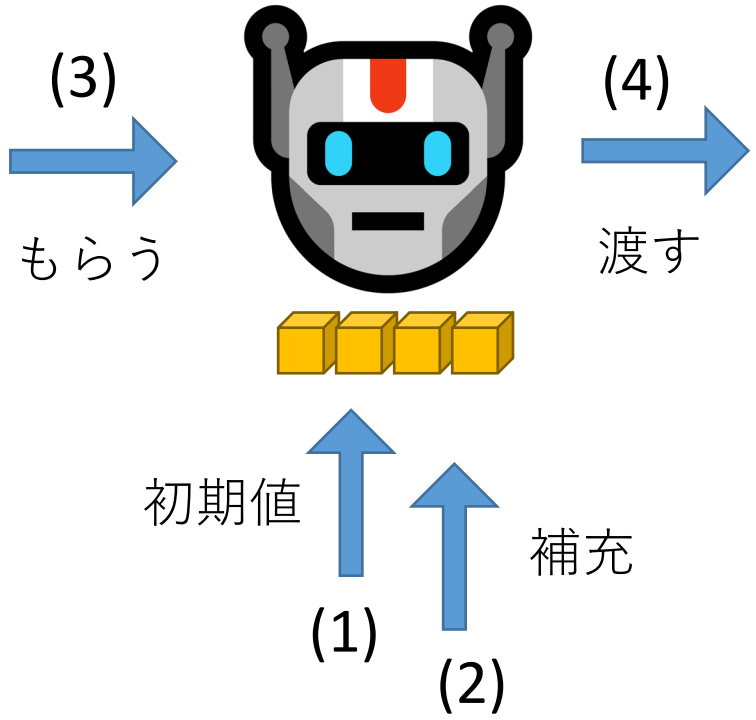
A_i (1)

＋(補充でもらう荷物の数の総和) (2)

＋(T時間までに左のロボットからもらう荷物の数) (3)

－(T時間までに右のロボットに渡す荷物の数) (4)

(1), (2)の計算は簡単である。(3)、(4)の値を考える



問題９　荷物渡し２

(3)の値

1体目のロボットがT時間までに左のロボットから貰う荷物の数 $= 0$　（左にロボットはいない）

$i(i > 1)$体目のロボットがT時間までに左のロボットから貰う荷物の数
$=$
$i - 1$体目のロボットがT時間までに右のロボットに渡す荷物の数
→　$i - 1$体目のロボットに関する(4)の値　→　各ロボットについて(4)の値が求まれば良い

(4)の値

ロボットはどのようなときに荷物を渡す？

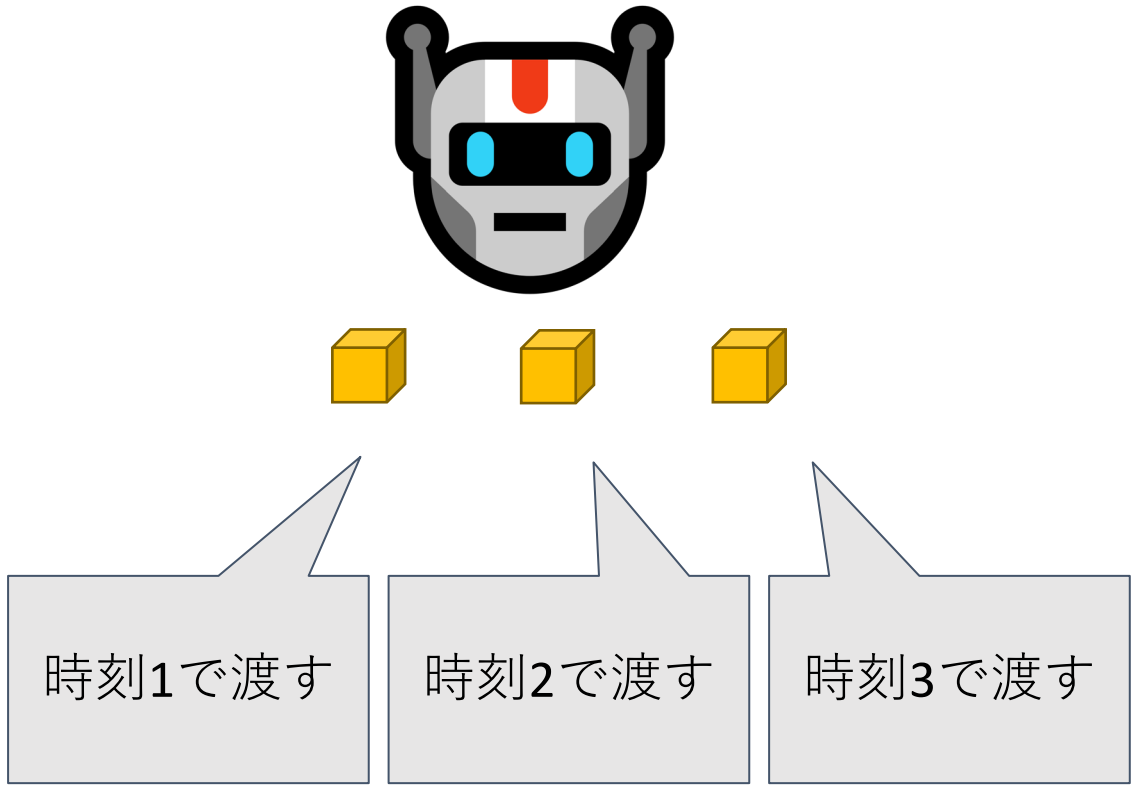
→　荷物を1個以上持っているとき
→　N体目のロボットだけ例外で、荷物を渡さない

i体目のロボットが持っている荷物の数が非零であるような操作の回数を求めればよい

問題９　荷物渡し２

考察：　渡す荷物を決める

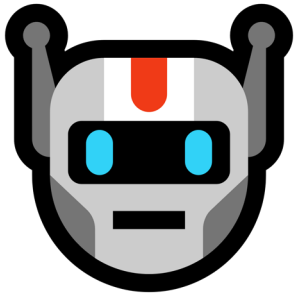
持っている荷物/補充される荷物をすべて区別することにして、「この荷物はこの時間に渡す」というのを各々決める



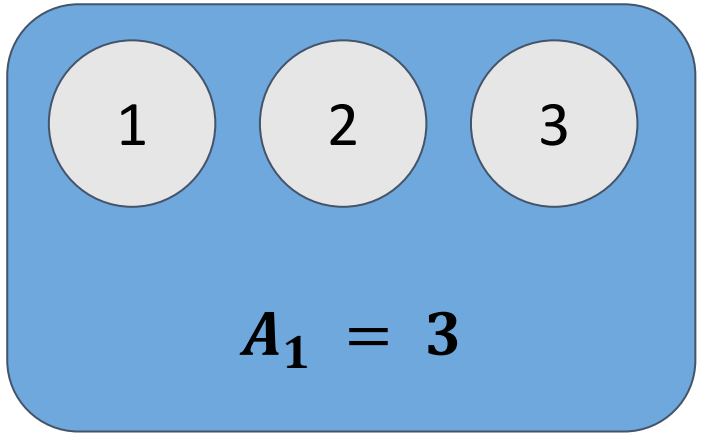
問題９　荷物渡し２

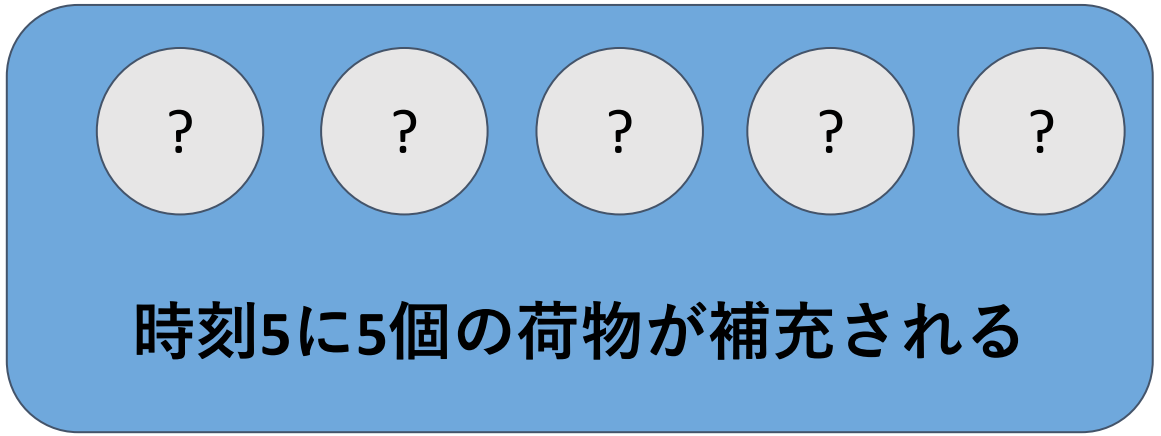
1体目のロボットについて

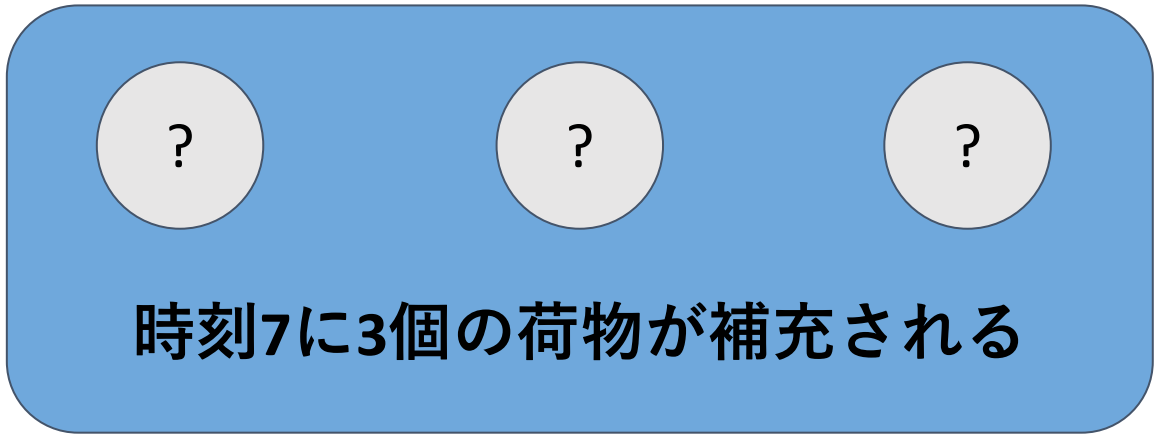
渡す時刻



補充される荷物はいつ渡せばいいんだろう？



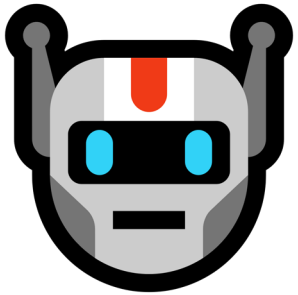




問題９　荷物渡し２

1体目のロボットについて

渡す時刻



時刻5に補充される荷物は時刻6以降に渡す

1　2　3

$A_1 = 3$

6　7　8　9　10

時刻5に5個の荷物が補充される

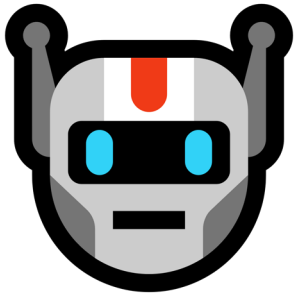
?　?　?

時刻7に3個の荷物が補充される

問題９　荷物渡し２

1体目のロボットについて

渡す時刻



時刻7に補充される荷物は時刻8以降に渡す…

1　2　3

$A_1 = 3$

6　7　8　9　10

時刻5に5個の荷物が補充される

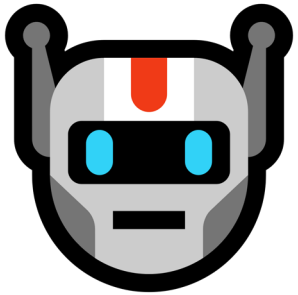
8　9　10

時刻7に3個の荷物が補充される

問題９　荷物渡し２

1体目のロボットについて

渡す時刻



重複が発生！

1　2　3

$A_1 = 3$

6　7　8　9　10

時刻5に5個の荷物が補充される

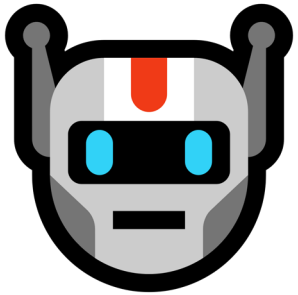
8　9　10

時刻7に3個の荷物が補充される

問題９　荷物渡し２

1体目のロボットについて

渡す時刻



つじつまを合わせる

1　2　3

$A_1 = 3$

6　7　8　9　10

時刻5に5個の荷物が補充される

11　12　13

時刻7に3個の荷物が補充される

問題９　荷物渡し２

1体目のロボットについて

荷物に番号をつける → **番号がつけられた荷物が存在する時刻の集合**を管理する

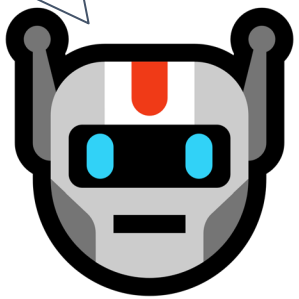
荷物が補充される → 追加される荷物について、$\boldsymbol{S}$ **+ 1以上の集合に存在しない最小の時刻**に荷物を渡すことにする＝集合にその値を追加する

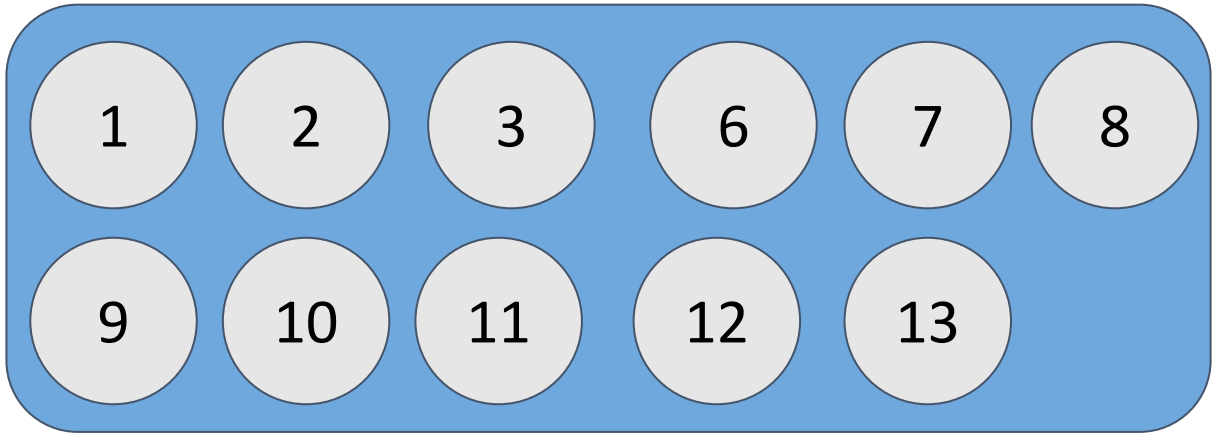
集合に存在するT以下の値の個数が解になる

問題９　荷物渡し２

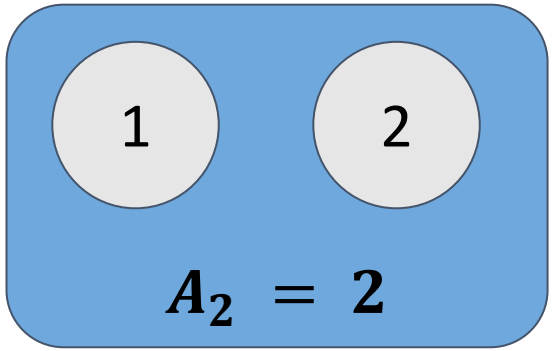
２体目のロボットについて

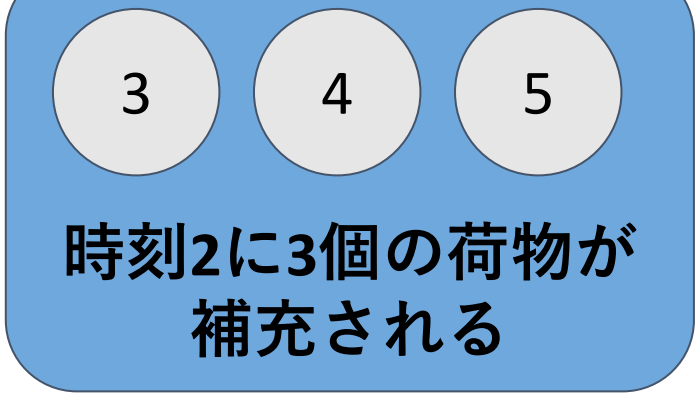
これらの時刻に荷物を渡します





貰う荷物はいつ渡せば良いのだろうか?

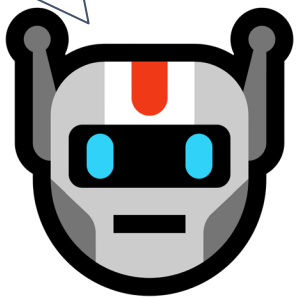




問題９　荷物渡し２

２体目のロボットについて

これらの時刻に荷物を渡します

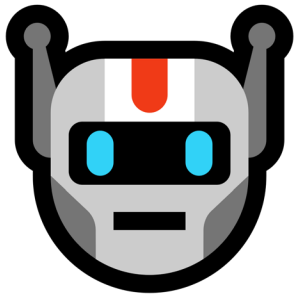


6　7　8　9　10　11

12　13　14　15　16

3　4　5

時刻2に3個の荷物が補充される



補充された荷物と同じ要領で渡す時刻を決める

1　2

$A_2 = 2$

問題９　荷物渡し２

２体目のロボットについて

$i-1$体目から貰う荷物それぞれについて、(貰う荷物の時刻をKとすると)、$K+1$以降の集合に属さない最小の時刻に荷物を渡すとしてよい

→　時刻Kに荷物が1個補充されると言い換えてよい

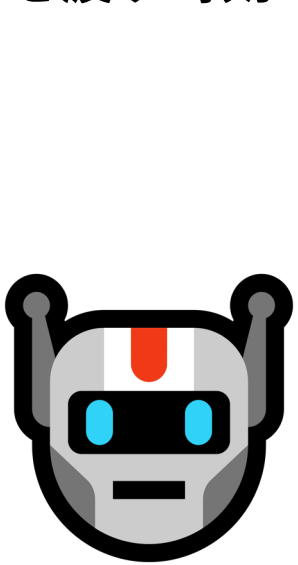
また、はじめから持っている荷物も「時刻0にA_i個の荷物が補充される」と言い換えると考察の見通しがよい

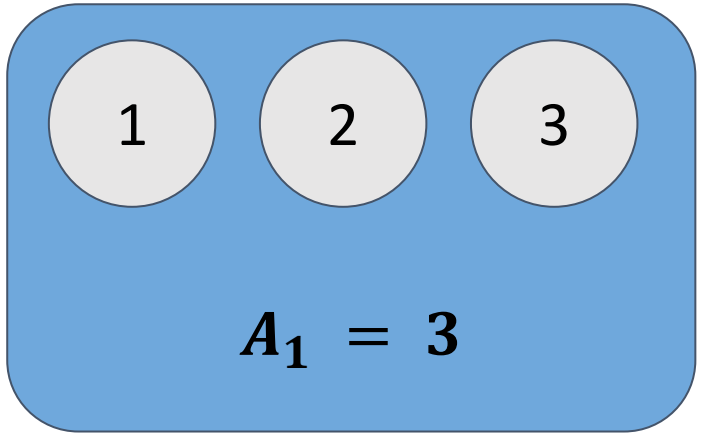
ここまでのアイデアで、(4)の値は左側のロボットから順番に計算できそうである。

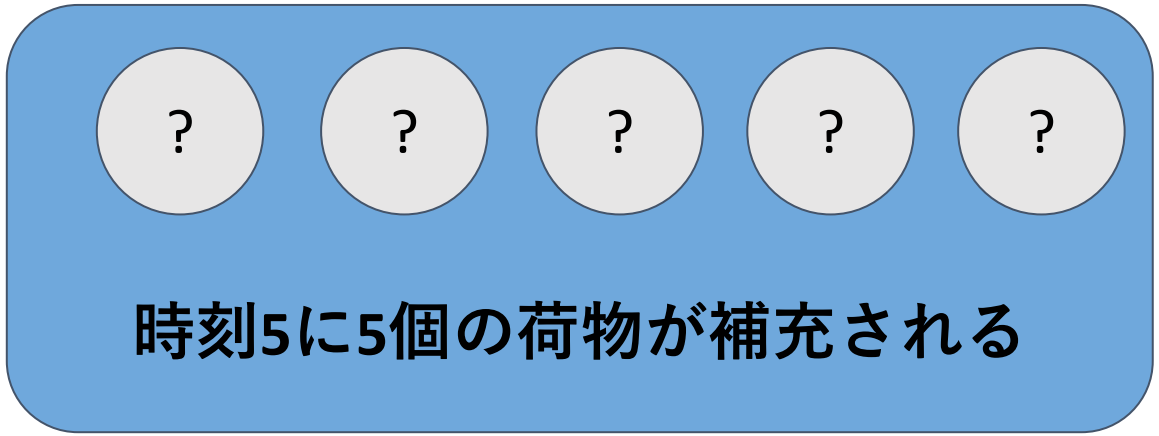
- しかし、補充される荷物の数は非常に大きいため、愚直にこれを実装すると時間制限
- もう一工夫必要

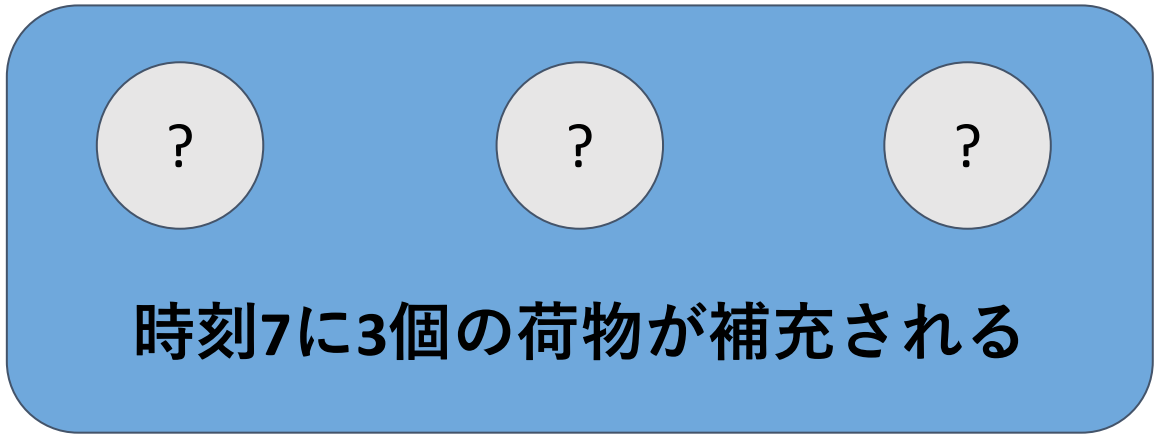
問題９　荷物渡し２

考察２：荷物を渡す時刻の集合





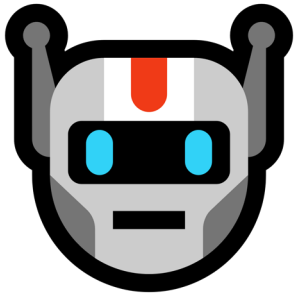




問題９　荷物渡し２

考察２：荷物を渡す時刻の区間の集合

区間で表現する



[1, 4)

$A_1 = 3$

? ? ? ? ?

時刻5に5個の荷物が補充される

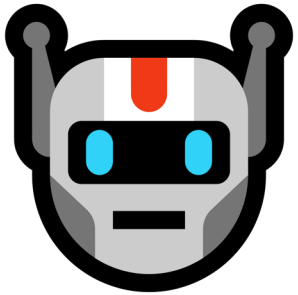
? ? ?

時刻7に3個の荷物が補充される

問題９　荷物渡し２

考察２：荷物を渡す時刻の区間の集合

追加される荷物を渡す時刻も区間をなす



[1, 4)

$A_1 = 3$

[6,11)

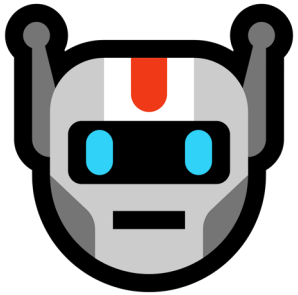
時刻5に5個の荷物が補充される

[11, 14)

時刻7に3個の荷物が補充される

問題９　荷物渡し２

考察２：荷物を渡す時刻の区間の集合



つなげられる区間はつなげてしまう

[1, 4)

$A_1 = 3$

[11, 14)

時刻5に5個の荷物が補充される

時刻7に3個の荷物が補充される

問題９　荷物渡し２

考察２：区間による荷物を渡す時刻の集合の表現

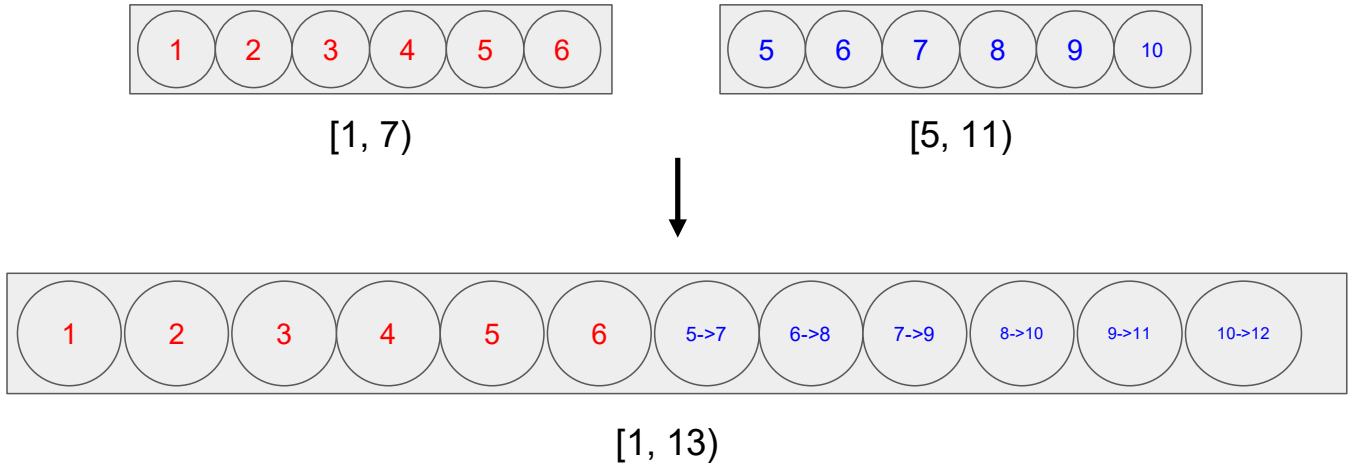
補充される荷物を渡す時刻の集合は1個以上の「時刻の区間」で表現することができる。

新しく時刻SからB個の荷物を追加するとき…

- 半開区間[S + 1, S + B + 1)の追加を試みる
- 集合内の共通区間(端点のみでもよい)を持つ区間すべてをマージする

[l1, r1)(集合内にある)と[l2, r2)(今追加しようとしている)のマージ

1. 集合から[l1, r1)を削除する
2. [l2, r2) ← [min(l1, l2), max(r1, r2) + 共通区間の長さ)とする



問題９　荷物渡し２

計算量

集合に追加する区間の数... ちょうど$N + M$個

集合に一個の区間を追加するとき

- 追加する区間の数はちょうど1個
- 削除される区間の数は、集合内にある要素の数で抑えられる
  - 追加された区間は高々1回削除されると言い換えることができる

平衡二分木(std::setなどの標準ライブラリで事足りる)でこれを管理

→　全体$O((N + M)\log(N + M)$で実現可能

問題９　荷物渡し２

実装方針

区間をsetで管理する

注意点1

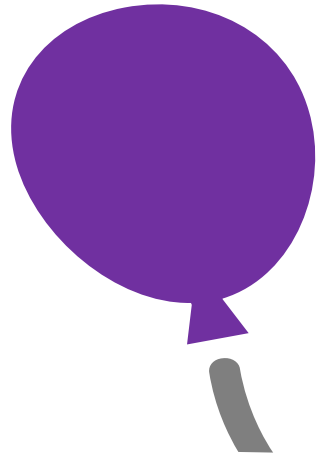
- 左のロボットから渡される荷物も「区間の挿入」として扱うが、これを愚直にやると計算量が壊れる
- $i-1$体目のロボットに関する区間の集合全体を1正側に平行移動する必要がある
  - 集合全体が平行移動した量を管理することで、みかけだけ動かす

注意点2

- マージによって、区間が[min(l1, l2), max(r1, r2) + 共通区間の長さ)と変化することを失念しないようにする

問題１０　学習データセットの分析（１２点）

正答数:　　５チーム



問題１０　学習データセットの分析

概要

- 長さ N の数列 A が与えられる
- 以下のクエリを Q 回処理せよ
  - $L\ R\ X$
    - $L \leq l \leq r \leq R$ を満たす (l, r) のうち、部分列 $A_l, A_{l+1}, \cdots, A_r$ に含まれる異なる整数の個数が X 個になるようなものは何通りか

制約

- $1 \leq N \leq 100000$
- $1 \leq Q \leq 100000$
- $1 \leq A_i \leq N$
- $1 \leq L_i \leq R_i \leq N$
- $1 \leq X_i \leq \min(N, 100)$

問題１０　学習データセットの分析

基本方針

元の問題

- $f(L,R,X) := L \leq l \leq r \leq R$ を満たす (l,r)のうち、部分列 $A_l, A_{l+1}, \cdots, A_r$に含まれる異なる整数の個数がちょうど X 個になるようなものは何通りか

新しい問題

- $g(L,R,X) := L \leq l \leq r \leq R$ を満たす (l,r) のうち、部分列 $A_l, A_{l+1}, \cdots, A_r$に含まれる異なる整数の個数が X 個以上になるようなものは何通りか

$f(L,R,X) = g(L,R,X) - g(L,R,X+1)$ と表せる

- 以降 $g(L,R,X)$ のみ考える

問題１０　学習データセットの分析

素朴な解法１（時間制限）

(l, r) を全探索する

- $O(QN^3)$ や $O(QN^2)$

高速化するには？

- l を固定したときに条件を満たす r の個数を高速に求めたい

```
long long g(int L, int R, int X) {
    long long ans = 0;
    for (int l = L; l <= R; l++)
        for (int r = l; r <= R; r++)
            if (A_l, ..., A_rが条件を満たす)
                ans += 1;
    return ans;
}
```

問題１０　学習データセットの分析

前計算

以下の値を前計算しておく

- $B[x][l]$: $A_l, A_{l+1}, \cdots, A_r$ に含まれる異なる整数の個数が X 個以上であるような最小の r. 存在しないならば $N+1$.

$B[x][l] \leq r \leq R$ を満たす r が条件を満たす

- 個数は $\max(R+1-B[x][l], 0)$ 個

問題１０　学習データセットの分析

前計算

以下の値を前計算しておく

- $B[x][l]: A_l, A_{l+1}, \cdots, A_r$に含まれる異なる整数の個数が X 個以上であるような最小の r. 存在しないならば $N+1$.

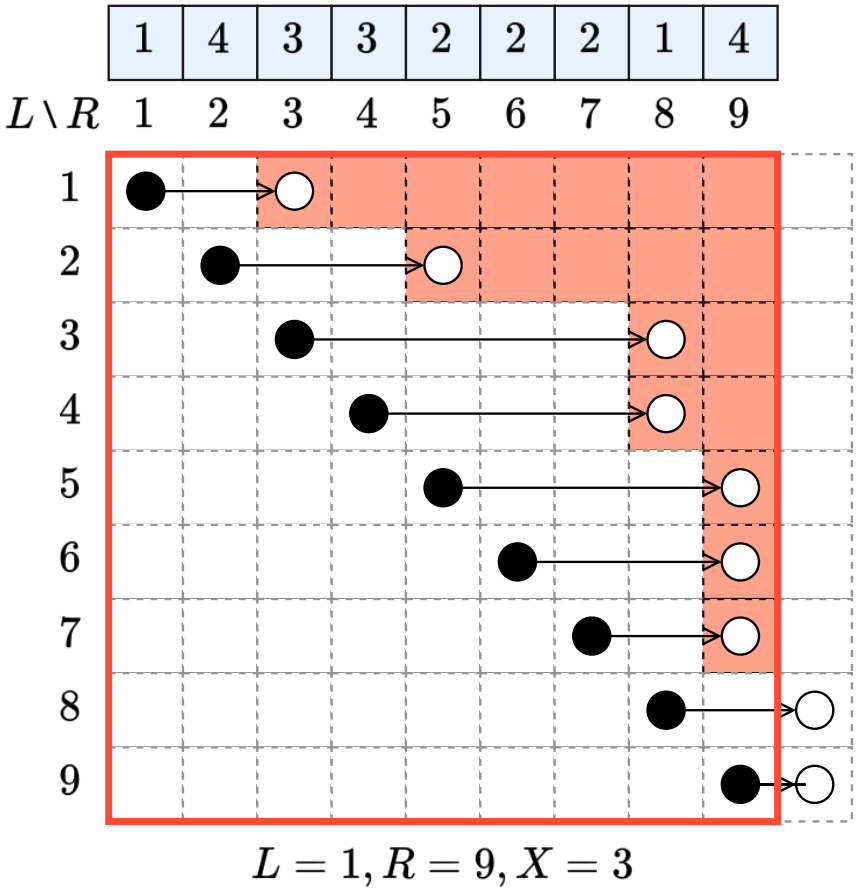
A	1	4	3	3	2	2	2	1	4
$B[3]$	3	5	8	8	9	9	9	10	10

問題１０　学習データセットの分析

前計算

$B[x][l] \leq r \leq R$ を満たす r が条件を満たす

- 個数は $\max(R+1-B[x][l], 0)$ 個



問題１０　学習データセットの分析

前計算

Bの特徴

- $B[x][l] \leq B[x][l+1]$

行ごとに**尺取法**を適用すると, 1行当たり $O(N)$ で計算できる

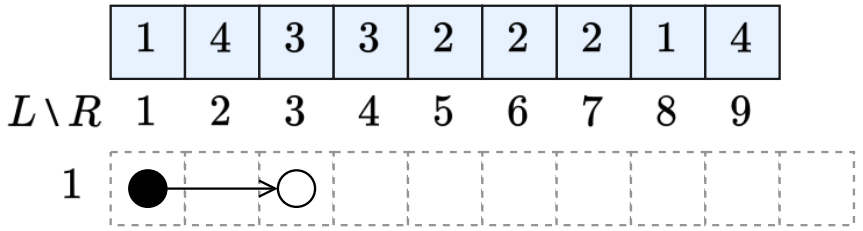
- 前計算 $O(N \max(X))$

A	1	4	3	3	2	2	2	1	4
$B[3]$	3	5	8	8	9	9	9	10	10

問題１０　学習データセットの分析

尺取法 X=3 (1 / 4)

$l = 1, r = 0$ として処理を開始



$r = 3$ まで伸ばすと $\{1, 3, 4\}$ の 3 個を含む

- $B[1] = 3$

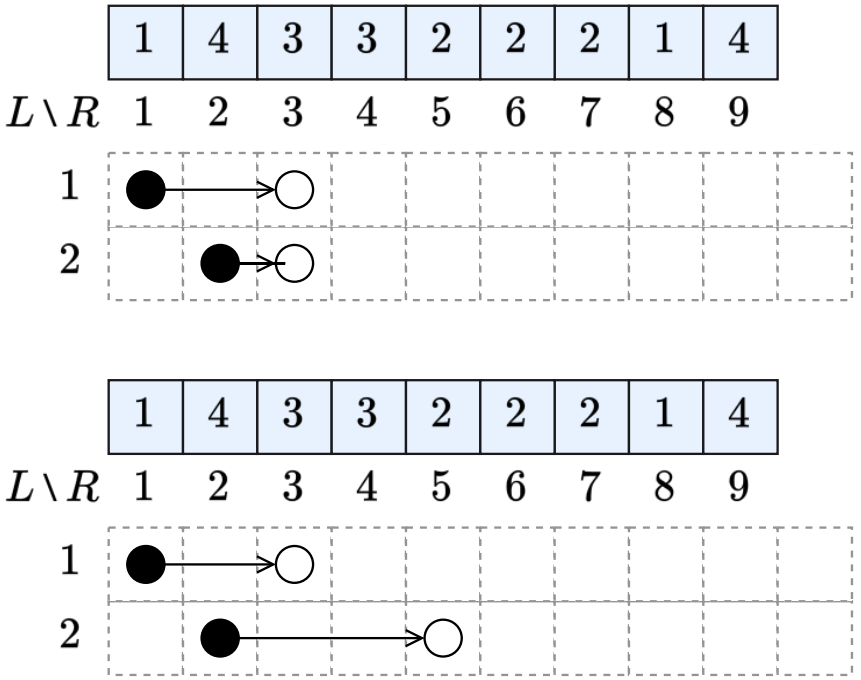
問題１０　学習データセットの分析

尺取法 X=3 (2 / 4)

$l = 2, r = 3$ として処理を継続

$r = 5$ まで伸ばすと $\{2, 3, 4\}$ の 3 個を含む

- $B[2] = 5$



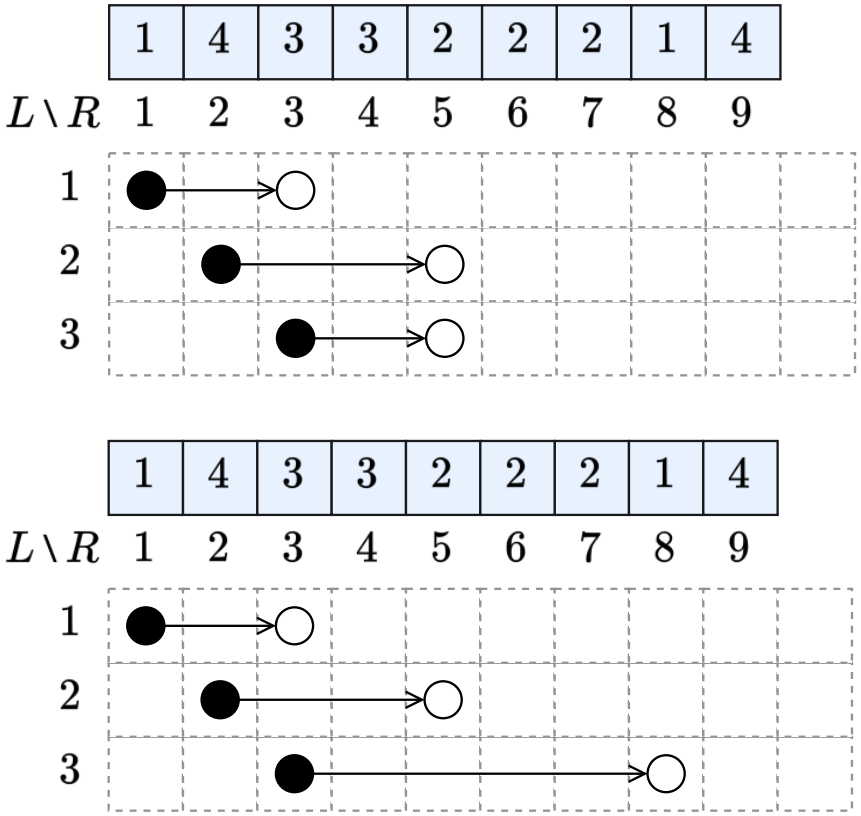
問題１０　学習データセットの分析

尺取法 X=3 (3 / 4)

$l = 3, r = 5$ として処理を継続

$r = 8$ まで伸ばすと $\{1, 2, 3\}$ の 3 個を含む

- $B[3] = 8$

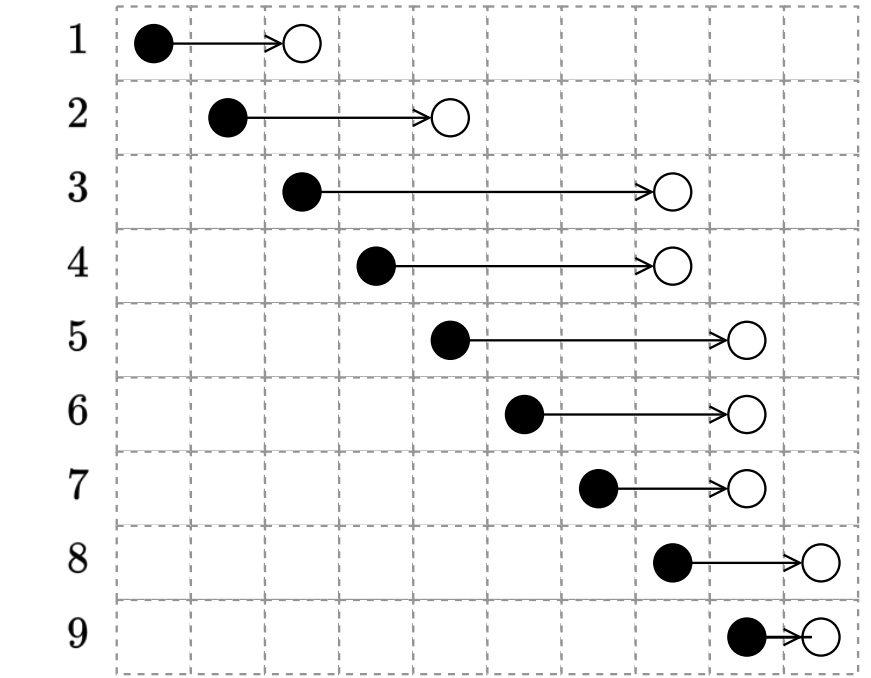


問題１０　学習データセットの分析

尺取法 X=3 (4 / 4)

最終的に右図のようになる

A	1	4	3	3	2	2	2	1	4
$B[3]$	3	5	8	8	9	9	9	10	10



$X = 3$

問題１０　学習データセットの分析

再掲

- $g(L,R,X) := L \leq l \leq r \leq R$ を満たす (l,r) のうち、部分列 $A_l, A_{l+1}, \cdots, A_r$ に含まれる異なる整数の個数が X 個以上になるようなものは何通りか

l を固定したときに条件を満たす r の個数は？

- $B[x][l] \leq r \leq R$ を満たす r が条件を満たす
- 個数は $\max(R+1-B[x][l], 0)$個

問題１０　学習データセットの分析

素朴な解法２（時間制限）

$L \leq l \leq R$ を満たす l に対して, $\max(R + 1 - B[x][l], 0)$ の総和を求めれば良い

- クエリあたり $O(N)$
- 前計算含めて全体で $O(N \max(X) + QN)$

総和の計算を高速化するには？

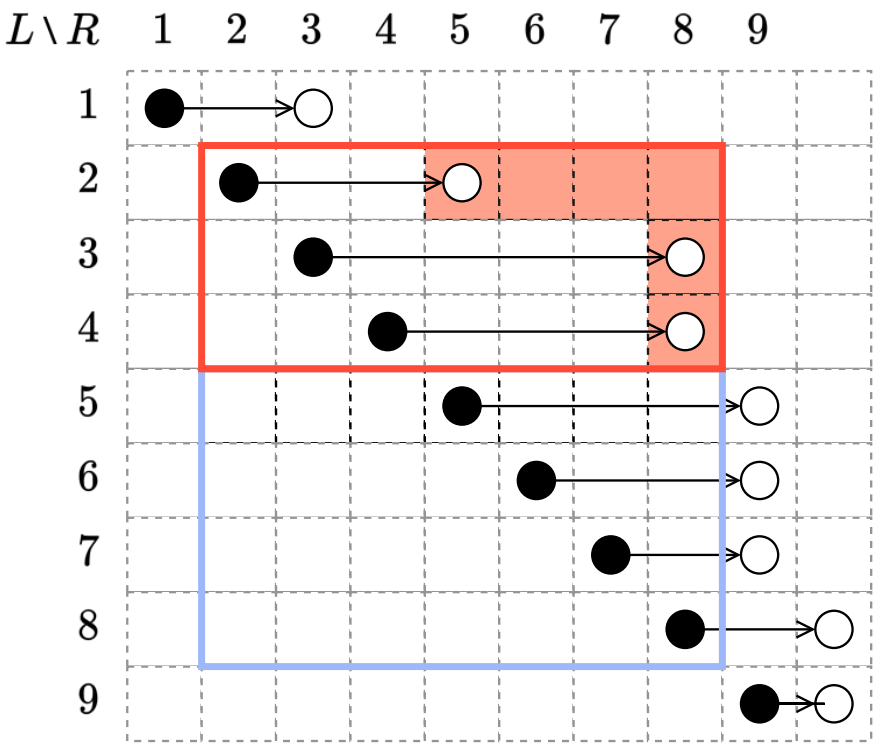
```
long long g(L, R, X) {
    long long ans = 0;
    for (int l = L; l <= R; l++)
        ans += max(R + 1 - B[x][l], 0);
    return ans;
}
```

問題１０　学習データセットの分析

想定解

- あるlについて、条件を満たすrの個数は$\max(R+1-B[x][l],0)$個
- $R+1 \leq B[x][l]$を満たすlの、解への寄与は0
- そのようなlを除外すると max を取り除ける
- $R+1 \leq B[x][l]$を満たす最小のlをR'とする
- $B[x][l] \leq B[x][l+1]$よりR'は**二分探索**で求められる

A	1	4	3	3	2	2	2	1	4
$B[3]$	3	5	8	8	9	9	9	10	10



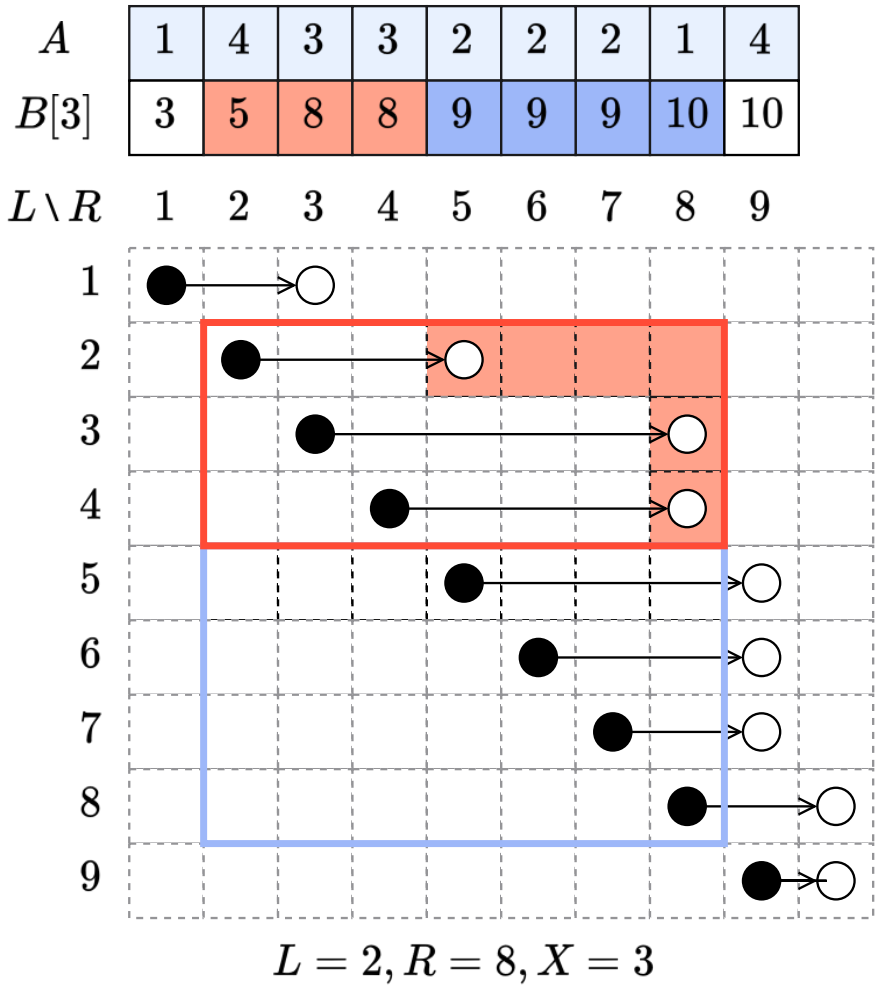
$L=2, R=8, X=3$

問題１０　学習データセットの分析

想定解

$L \leq l < R'$ を満たす l に対して, $R+1 - B[x][l]$ の総和を求めたい

- $R+1$ の総和から $B[x][l]$ の総和を引けば良い
- 前者は $(R+1)(R'-L)$
- 後者は B の各行に対する**累積和**を求めておけば高速に計算できる



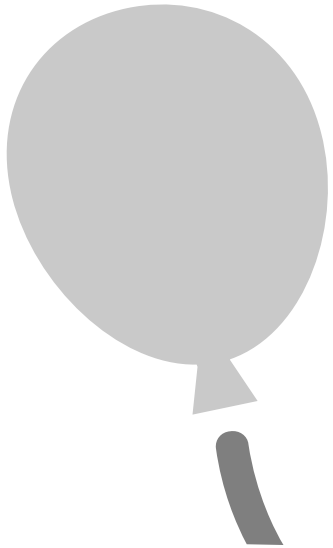
問題１０　学習データセットの分析

計算量

- 前計算 $O(N \max(X))$
- クエリあたり $O(\log N)$
- 全体 $O(N \max(X) + Q\log N)$

問題１１　六角沼からの脱出（１２点）

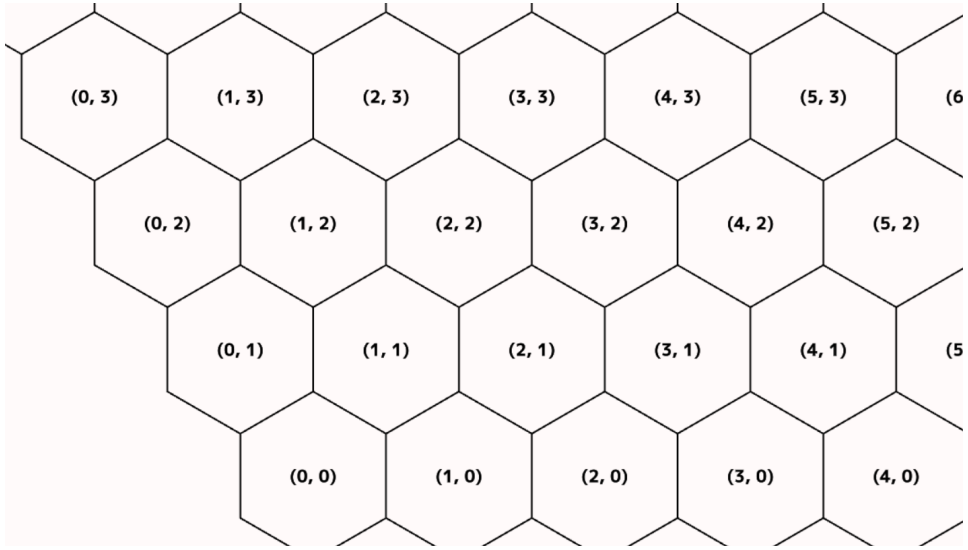
正答数:　　１チーム



問題１１　六角沼からの脱出

概要

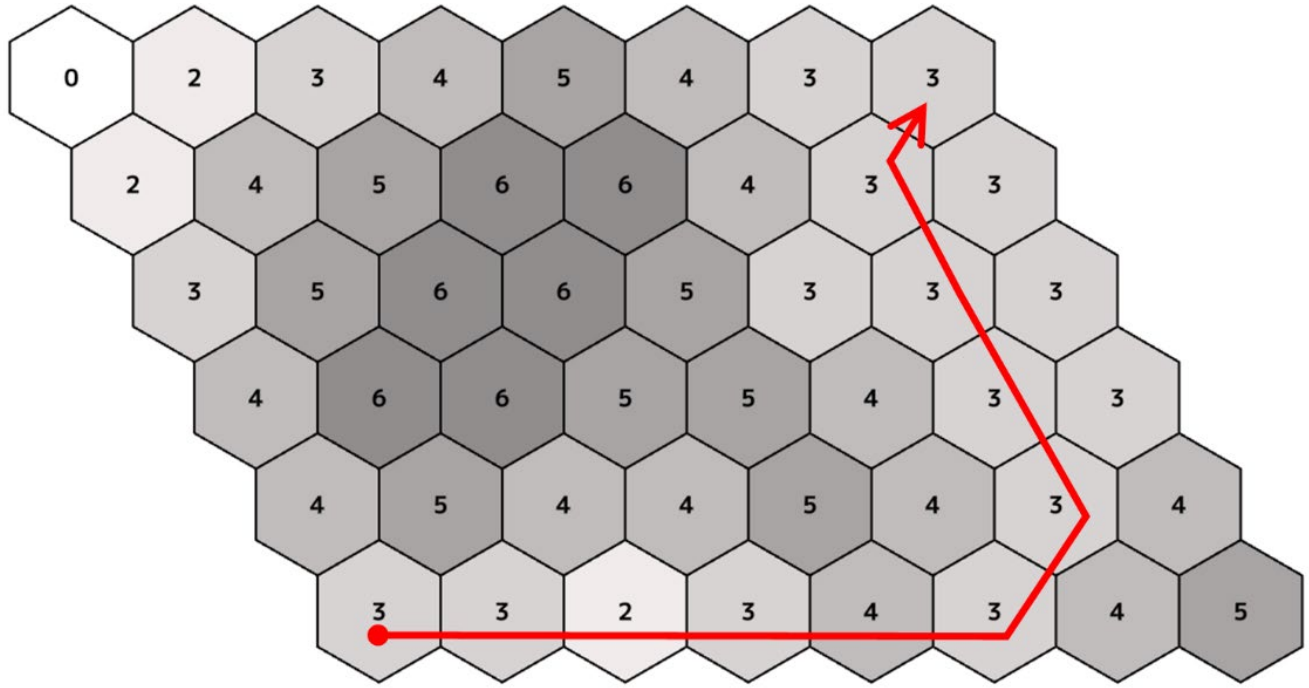
- $(H+1) \times (W+1)$個の正六角形が平面上に並んでいる。
- 各マスに危険度という指標を定める。はじめは、全ての六角形の危険度は0である。
- $i = 1, 2, \ldots, N$について、マス(x_i, y_i)にa_iリットルの毒を散布する
- それに応じて、マス(x_i, y_i)からの最短距離が$d(d = 0, 1, 2, \ldots)$であるマスの危険度が$\max\{a_i - d,\ 0\}$増加する。
- マス$(0, 0)$からマス(W, H)へ移動するとき、通りうるマスの最大値としてありうる最小値を求めよ。



問題１１　六角沼からの脱出

制約

- $1 \leq H, W \leq 200000$
- $1 \leq H \times W \leq 200000$
- $1 \leq N \leq 2 \times 100000$
- $1 \leq a_i \leq 10^9$



入出力例２の答えは4

問題１１　六角沼からの脱出

方針

$1 \leq H \times W \leq 2 \times 10^5$という制約より、マスの数は抑えられている

💡すべてのマスの危険度をあらかじめ求めることができたならば、解も求められるように感じられる

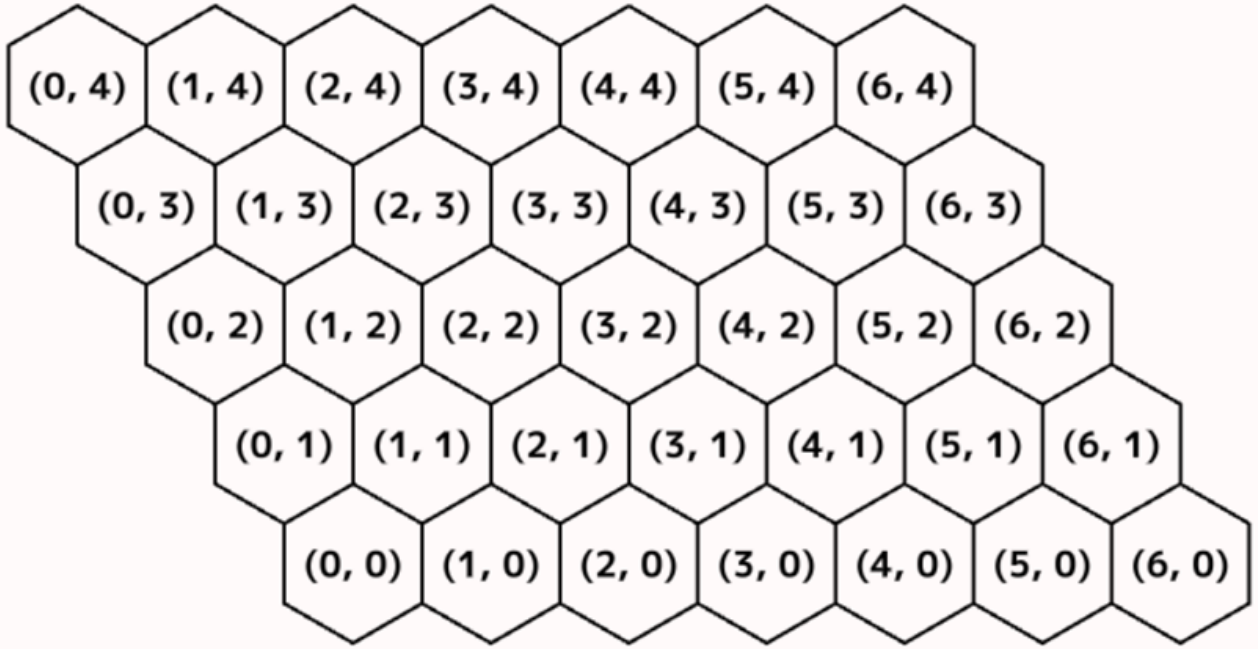
- 様々な解法がある
- 詳しくは後述する

問題１１　六角沼からの脱出

考察

六角形座標は扱いづらいので、二次元グリッドに直してみる。

マス(x, y)を左からx列目、下からy行目の格子に割り当てる。



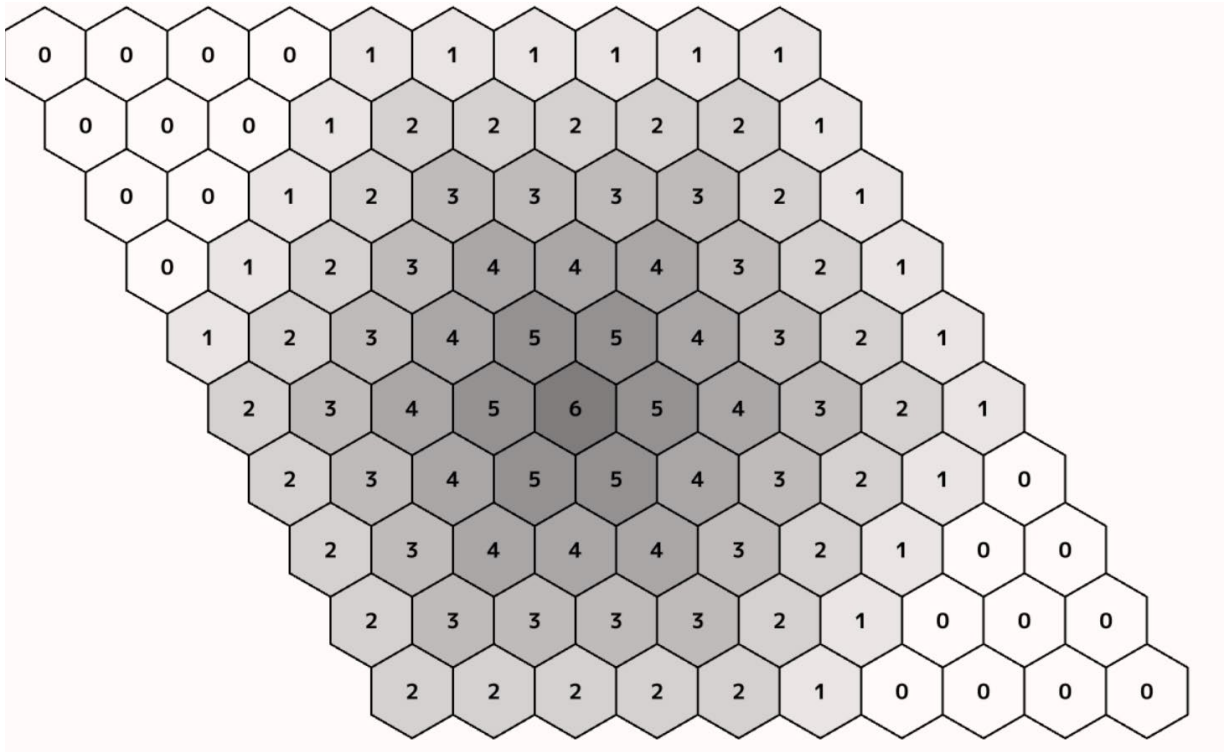
(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)
(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	(6, 3)
(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)
(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)
(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)	(5, 0)	(6, 0)

問題１１　六角沼からの脱出

考察

毒の散布をこのグリッド上で描画してみると....

良い特徴・規則性を探してみよう



0	0	0	0	1	1	1	1	1	1
0	0	0	1	2	2	2	2	2	1
0	0	1	2	3	3	3	3	2	1
0	1	2	3	4	4	4	3	2	1
1	2	3	4	5	5	4	3	2	1
2	3	4	5	6	5	4	3	2	1
2	3	4	5	5	4	3	2	1	0
2	3	4	4	4	3	2	1	0	0
2	3	3	3	3	2	1	0	0	0
2	2	2	2	2	1	0	0	0	0

問題１１　六角沼からの脱出

考察

ある行について、毒の散布を行ったときの危険度の上昇具合を観察する

- はじめに+1, +2, +3, …., と危険度の上昇度合いが1ずつ上昇していく (1)
- つぎに、+c, +c, +c, ….と危険度の上昇度合いが一定になる (2)
- 最後に、+c - 1, +c - 2, …, +1と危険度の上昇度合いが1ずつ減っていく (3)

(1)　(2)　(3)

問題１１　六角沼からの脱出

危険度の加算を高速に

愚直に加算をシミュレートする → $O(HWN)$で時間制限

- 加算の仕方を工夫する必要がある

(1)　(2)　(3)

(1), (2), (3)の部分別々に、高速に危険度の加算を実現する方法を考える

問題１１　六角沼からの脱出

危険度の加算を高速に

(2)に関しては、累積和の手法を用いることで、加算しなければいけない箇所が減る（imos法）

0	+C	0	0	0	0	0	0	-C	0

累積和をとる

差分をとる

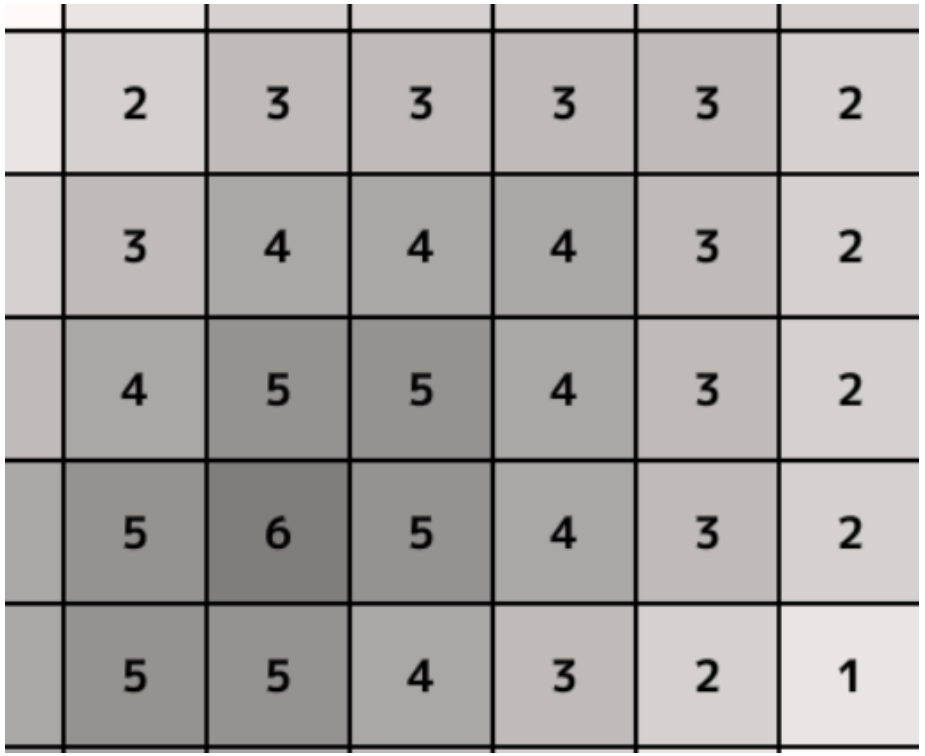
0	+C	+C	+C	+C	+C	+C	+C	0	0

問題１１　六角沼からの脱出

危険度の加算を高速に

(2)に関して、imos法を用いると一回の毒の散布で$O(N)$個の一点加算、最後にH回長さWの列に対して累積和を取る。→　全体で$O(HN + HW)$かかる

少し改善したが、まだ時間制限



	+3				-3
	+4			-4	
	+5		-5		
	+6	-6			
+5		-5			

問題１１　六角沼からの脱出

危険度の加算を高速に

(1), (3)は「公差1(-1)の等差数列」という形になっていた。このような数列に対して差分をとってみる

0	+1	+2	+3	+4	+5	+6	0	0	0

問題１１　六角沼からの脱出

0	+1	+2	+3	+4	+5	+6	0	0	0

差分をとる

0	+1	+1	+1	+1	+1	+1	-6	0	0

差分をとる

0	+1	0	0	0	0	0	-7	+6	0

問題１１　六角沼からの脱出

0	+1	+2	+3	+4	+5	+6	0	0	0

累積和をとる

0	+1	+1	+1	+1	+1	+1	-6	0	0

累積和をとる

0	+1	0	0	0	0	0	-7	+6	0

問題１１　六角沼からの脱出

(1), (3)もimos法で可能

累積和×２

+1	+2	+3	+4	…		+C			
+1							-(C + 1)	+C	

1. 1があるところに+1する
2. Cがあるところの一つ右に-(C + 1)する
3. Cがあるところの二つ右に+Cする
4. **二回**累積和をとる

ことで、(1, 2, 3, …., C)という危険度の加算度合いを表現することができる

- 累積和を取る方向などを工夫すると、同様の方法で(C, C - 1, …, 1)も作ることが可能

(2)と同様に、$O(HN + HW)$で(1), (3)の部分の加算も実現できるようになった

問題１１　六角沼からの脱出

計算量の改善

- ここまでの議論を踏まえると、時間計算量$O(HN + HW)$、空間計算量$O(HW)$ですべてのマスの危険度を計算することができる。
- 時間計算量に関してまだ改善が必要である。

問題１１　六角沼からの脱出

再び考察

0	0	0	0	1	1	1	1	1	1
0	0	0	1	2	2	2	2	2	1
0	0	1	2	3	3	3	3	2	1
0	1	2	3	4	4	4	3	2	1
1	2	3	4	5	5	4	3	2	1
2	3	4	5	6	5	4	3	2	1
2	3	4	5	5	4	3	2	1	0
2	3	4	4	4	3	2	1	0	0
2	3	3	3	3	2	1	0	0	0
2	2	2	2	2	1	0	0	0	0

実は、行では無く**列**に対して加算度合いを見ても同様の議論ができる！

- 1ずつ増えて
- 増えなくなって
- 1ずつ減る

1	1	1
2	2	2
3	3	2
4	3	2
4	3	2
4	3	2
3	2	1
2	1	0
1	0	0
0	0	0

問題１１　六角沼からの脱出

再び考察

列に対して同様の方法で危険度が計算できることがわかった。

つまり、O(WN + HW)時間でもすべてのマスの危険度を計算可能

- これでも依然時間制限
- O(HN + HW)とたいして変わらない？

単純なアイデアを導入してみる。

- $H \leq W$ならば $O(HN + HW)$ 解法を、そうでなければ $O(WN + HW)$ 解法を利用する
- 時間計算量は $O(\min\{H, W\}\, N + HW)$ になる
  - これは高速？

問題１１　六角沼からの脱出

$O(min\{H,W\}N + HW)$の解析

$$\min\{H, W\}^2 \leq HW$$

$$\Rightarrow \min\{H, W\} \leq \sqrt{HW}$$ が成り立つ

$$\sqrt{HW} \leq \sqrt{2 \times 10^5} < 448$$ であり、 $N \leq 2 \times 10^5$

であることも踏まえると時間実行制限に間に合わせるのに現実的な時間計算量になっている。

問題１１　六角沼からの脱出

解の計算

すべてのマスの危険度を計算することができた。各マスの危険度を利用して、例えば以下の方法のいずれかで解を計算することができる

二分探索をする。危険度x以下のマスのみを通ってマス$(0,0)$からマス(W,H)に到達できるか？という判定問題を解く。これはDFSで可能。危険度に関して予め座標圧縮を施すと$O(HWlogHW)$

危険度に関して昇順にマスを並べ、マス$(0,0)$からマス(W,H)が連結になるまで危険度の小さいマスから隣接するマスとマージしていく。素集合データ構造を使うことができて$O(HWlogHW)$(ソートがボトルネック)

$dp_{x,y}$:=マス $(0,0)$からマス(W,H)に到達するために通る必要のあるマスの危険度としてありうる最小値愚直にやると時間制限となるが、priority queueを利用すると高速化することができる。$O(HWlogHW)$

問題１２　倉庫管理ロボット　（１２点）

正答数:　　１チーム

問題１２　倉庫管理ロボット

概要

- N 個の空の箱が一列に並んでいる
- 以下の二種類の命令を Q 回処理せよ
  - $1\ l\ r\ w\ c$

    $l, l+1, \cdots, r$ 番目の箱それぞれに重さ w の荷物を c 個追加する
  - $2\ x\ k$

    箱 x に入っている荷物のうち軽い方から k 番目の荷物の重さを求める

制約

- $1 \leq N \leq 200000$
- $1 \leq Q \leq 200000$
- $1 \leq w_i \leq 10^9$
- $1 \leq c_i \leq 10000$

問題１２　倉庫管理ロボット

考察

どのように箱 x に含まれる荷物を管理するか？

どのように箱 x に含まれる荷物のうち軽い方から k 番目の荷物の重さを求めるか？

以下のクエリに高速に答えられたら、二分探索で求まる

箱 x に含まれる、重さ v 以下の荷物の個数は k 未満か？

問題１２　倉庫管理ロボット

想定解 二次元セグメント木上の二分探索

箱 x に含まれる重さ v 以下の荷物の個数は k 未満か？

- 箱の番号と重さを座標とし、荷物の個数を座標上の点の値とする二次元平面上の一点加算・矩形領域和問題に帰着できる

荷物追加クエリ $\mathbf{1\ l\ r\ w\ c}$

- 点 (l, w) に $+c$，　点 $(r\ +\ 1, w)$ に $-c$ する

求値クエリ $\mathbf{2\ x\ k}$

- 箱 x に含まれる重さ v 以下の荷物の個数 $= [1, x]\ \times\ [1, v]$ の総和

矩形領域和クエリ一回あたり $O(\log N \log Q)$

- 普通に二分探索をすると $O(\log N \log^2 Q)$
- 通常のセグメント木と同様に二分探索をすると $O(\log N \log Q)$

問題１２　倉庫管理ロボット

二次元セグメント木について(1 / 4)

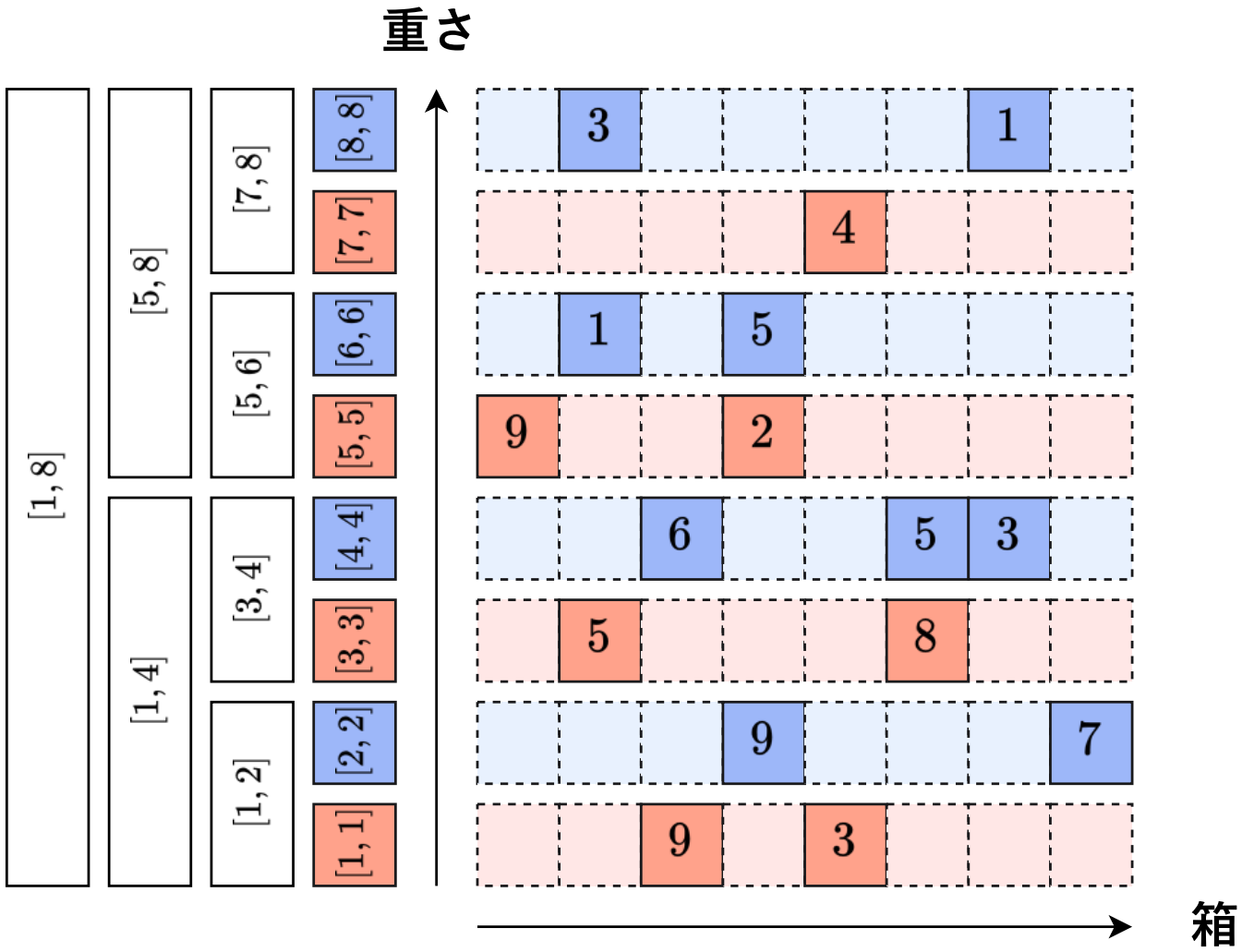
セグメント木の各ノードにセグメント木を乗せる

今回は座標値が大きい

- $N \leq 2 \times 10^5$
- $w_i \leq 10^9$

事前に一点加算クエリの座標を求めておいて圧縮する

- 連続する 1 行をセグメント木で管理する



問題１２　倉庫管理ロボット

二次元セグメント木について(2 / 4)

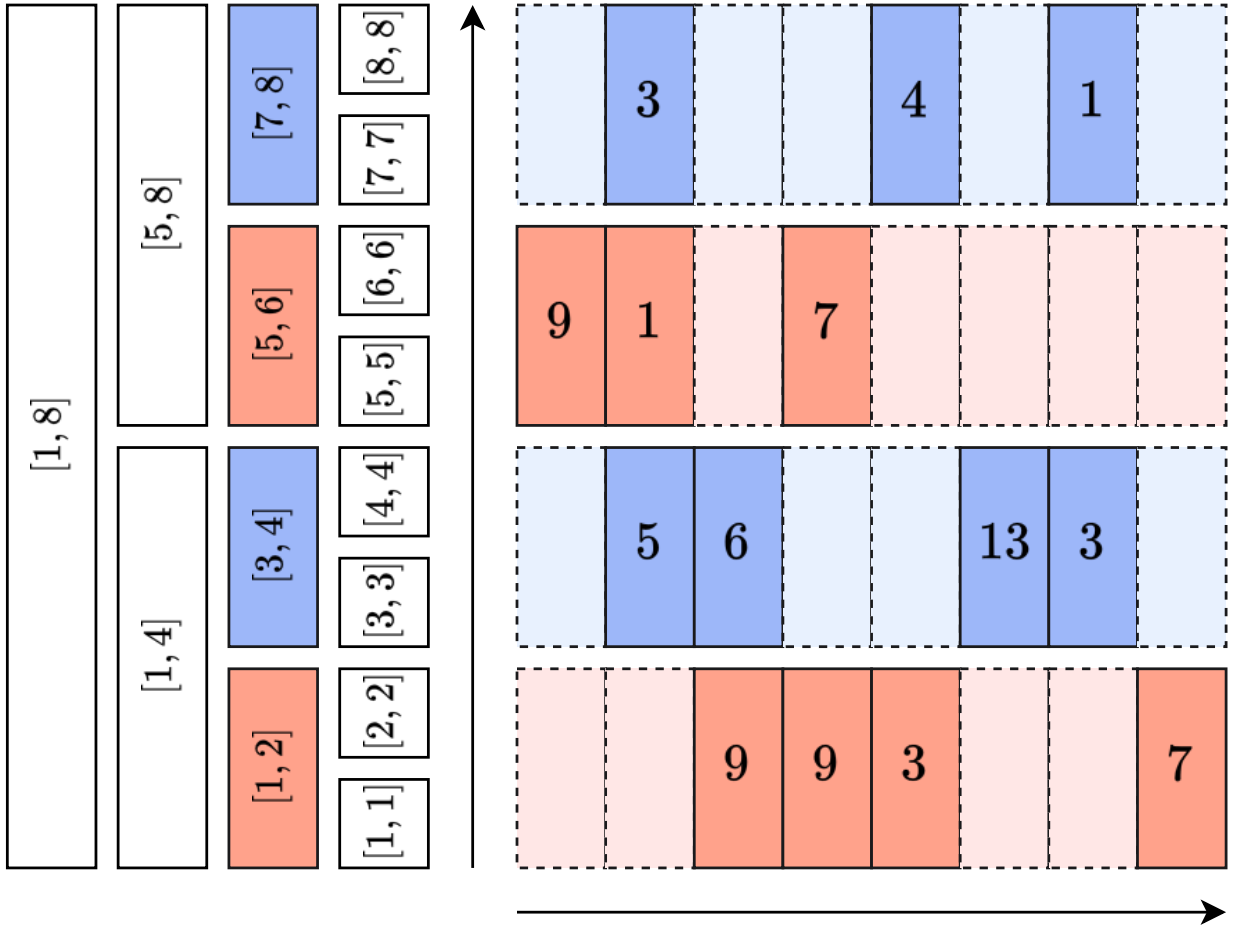
セグメント木の各ノードにセグメント木を乗せる

今回は座標値が大きい

- $N \leq 2 \times 10^5$
- $w_i \leq 10^9$

事前に一点加算クエリの座標を求めておいて圧縮する

- 連続する 2 行をセグメント木で管理する



問題１２　倉庫管理ロボット

二次元セグメント木について(3 / 4)

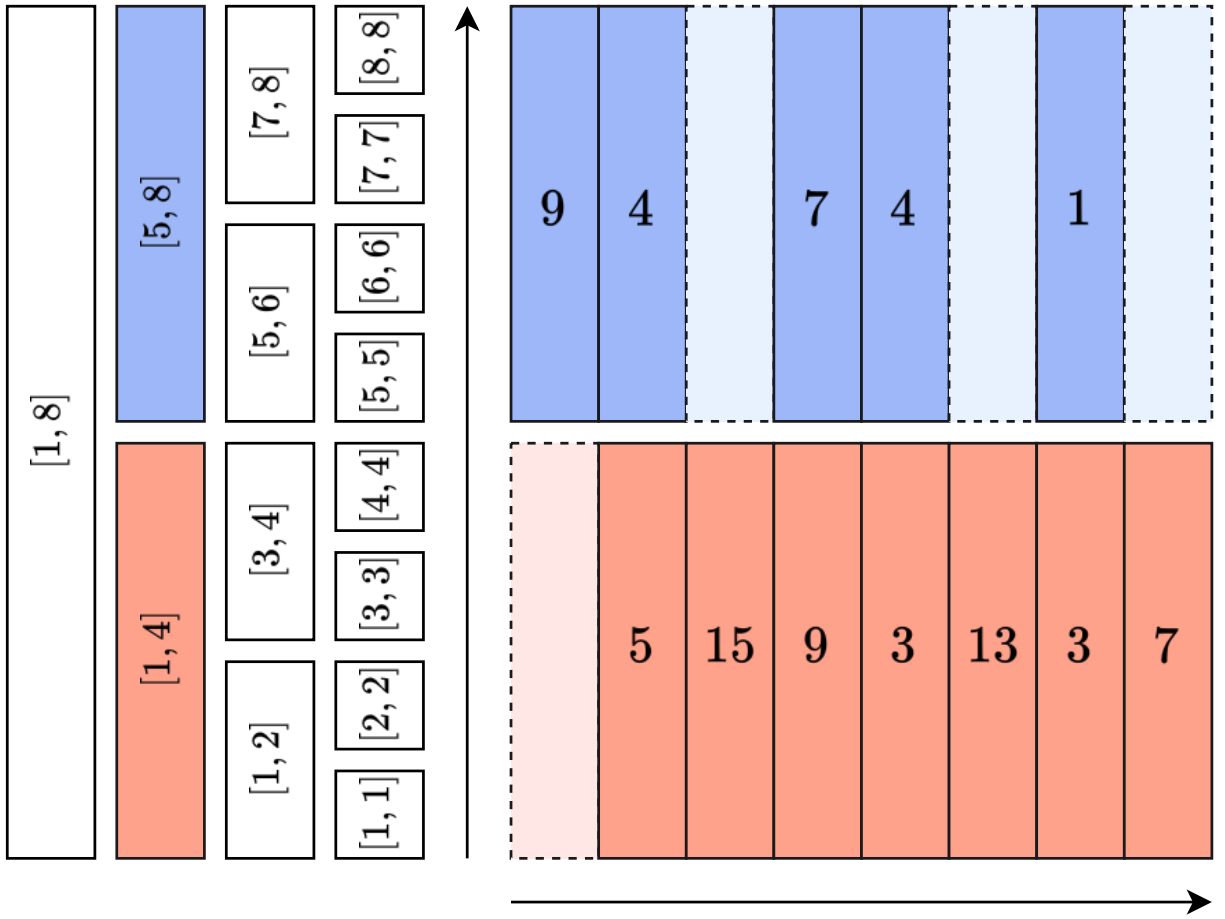
セグメント木の各ノードにセグメント木を乗せる

今回は座標値が大きい

- $N \leq 2 \times 10^5$
- $w_i \leq 10^9$

事前に一点加算クエリの座標を求めておいて圧縮する

- 連続する 4 行をセグメント木で管理する



問題１２　倉庫管理ロボット

二次元セグメント木について(4 / 4)

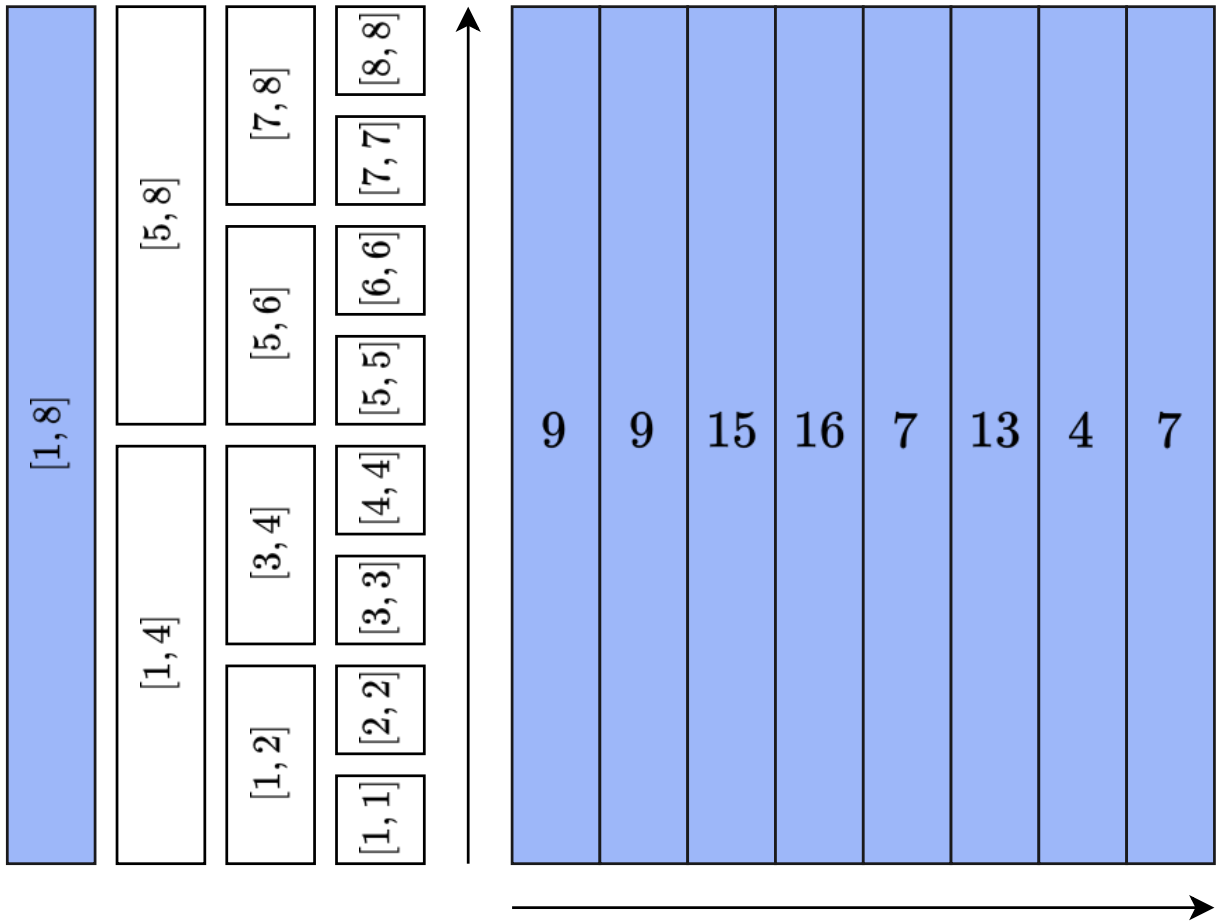
セグメント木の各ノードにセグメント木を乗せる

今回は座標値が大きい

- $N \leq 2 \times 10^5$
- $w_i \leq 10^9$

事前に一点加算クエリの座標を求めておいて圧縮する

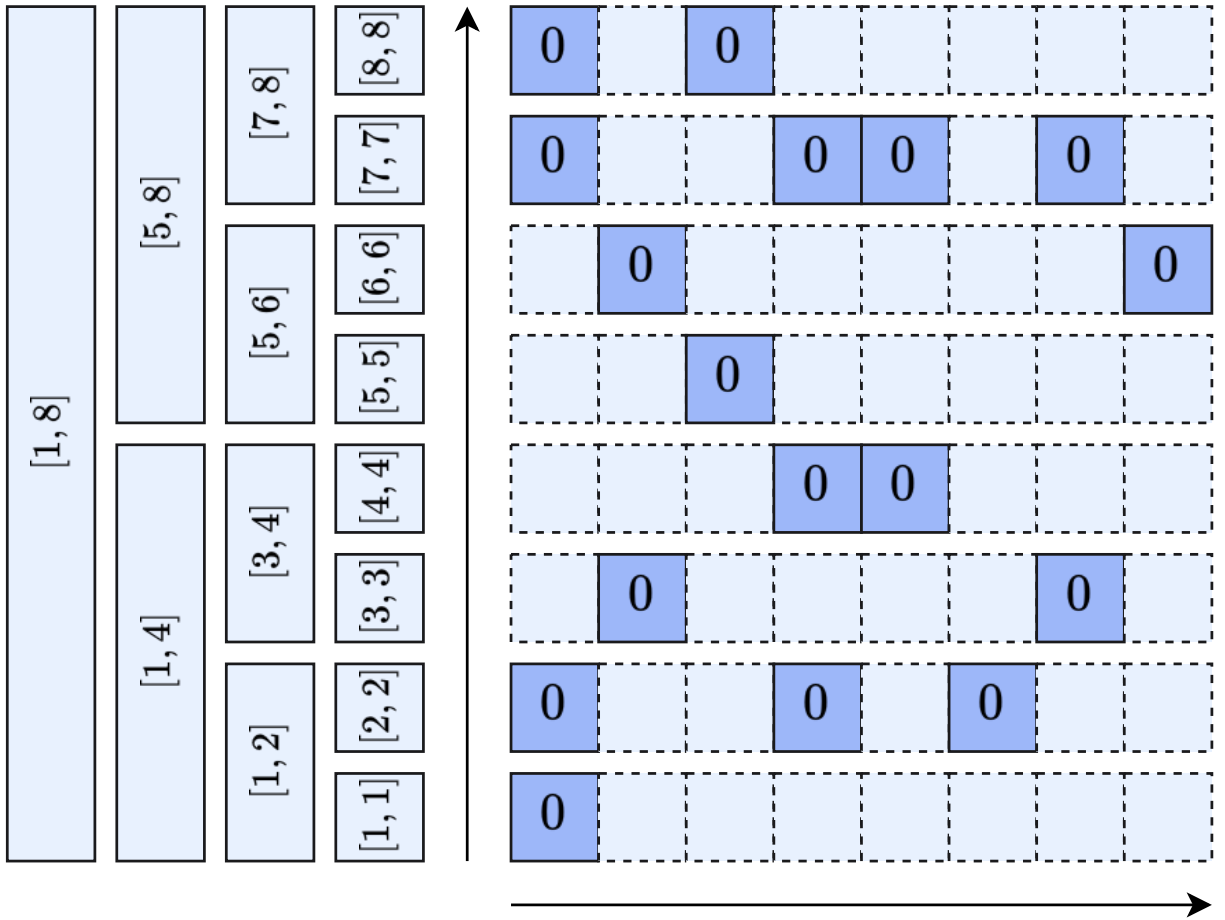
- 連続する 8 行をセグメント木で管理する



問題１２　倉庫管理ロボット

入力例

```
8 _
1 2 7 6 3
1 4 8 2 2
1 1 8 1 5
1 3 8 5 1
1 1 5 2 3
2 4 7
⋮
```



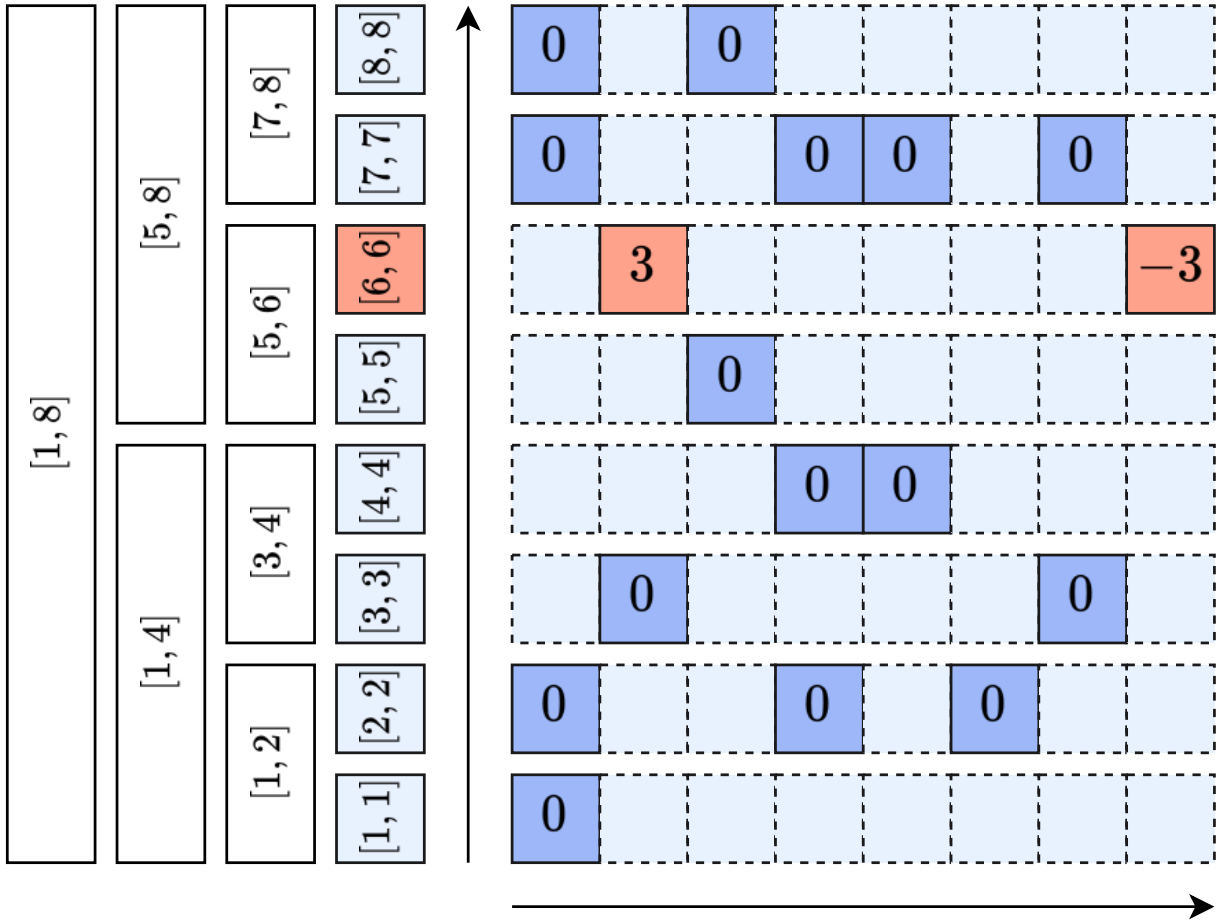
問題１２　倉庫管理ロボット

荷物追加クエリ(1 / 9)

$1\ 2\ 7\ 6\ 3$

- 箱 $2, 3, \cdots, 7$ に重さ 6 のボールを 3 個追加
- $(2, 6)$ に $+3$
- $(8, 6)$ に -3

根の方に移動しながら各ノードのセグメント木を更新していく



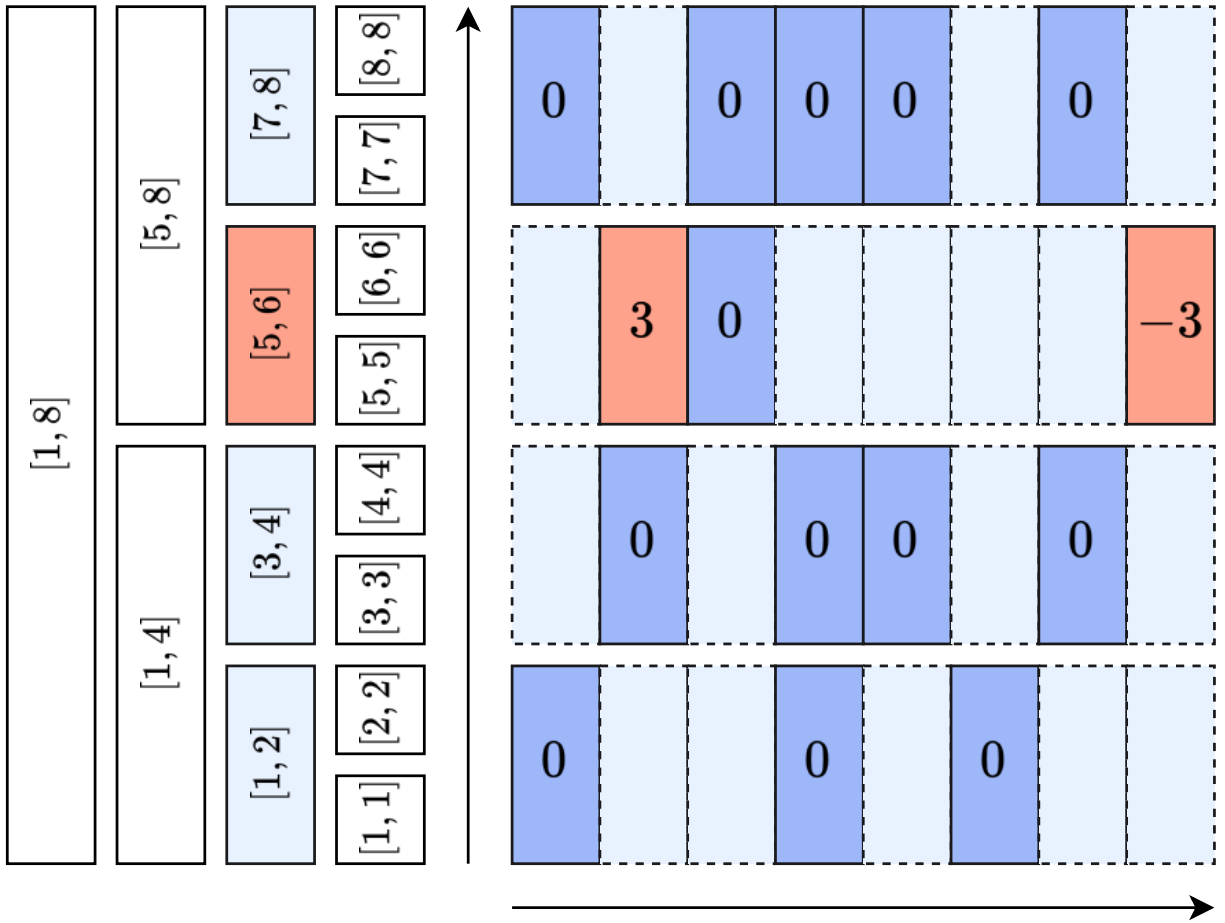
問題１２　倉庫管理ロボット

荷物追加クエリ(2 / 9)

1 2 7 6 3

- 箱 $2, 3, \cdots, 7$ に重さ 6 のボールを 3 個追加
- $(2, 6)$ に $+3$
- $(8, 6)$ に -3

根の方に移動しながら各ノードのセグメント木を更新していく



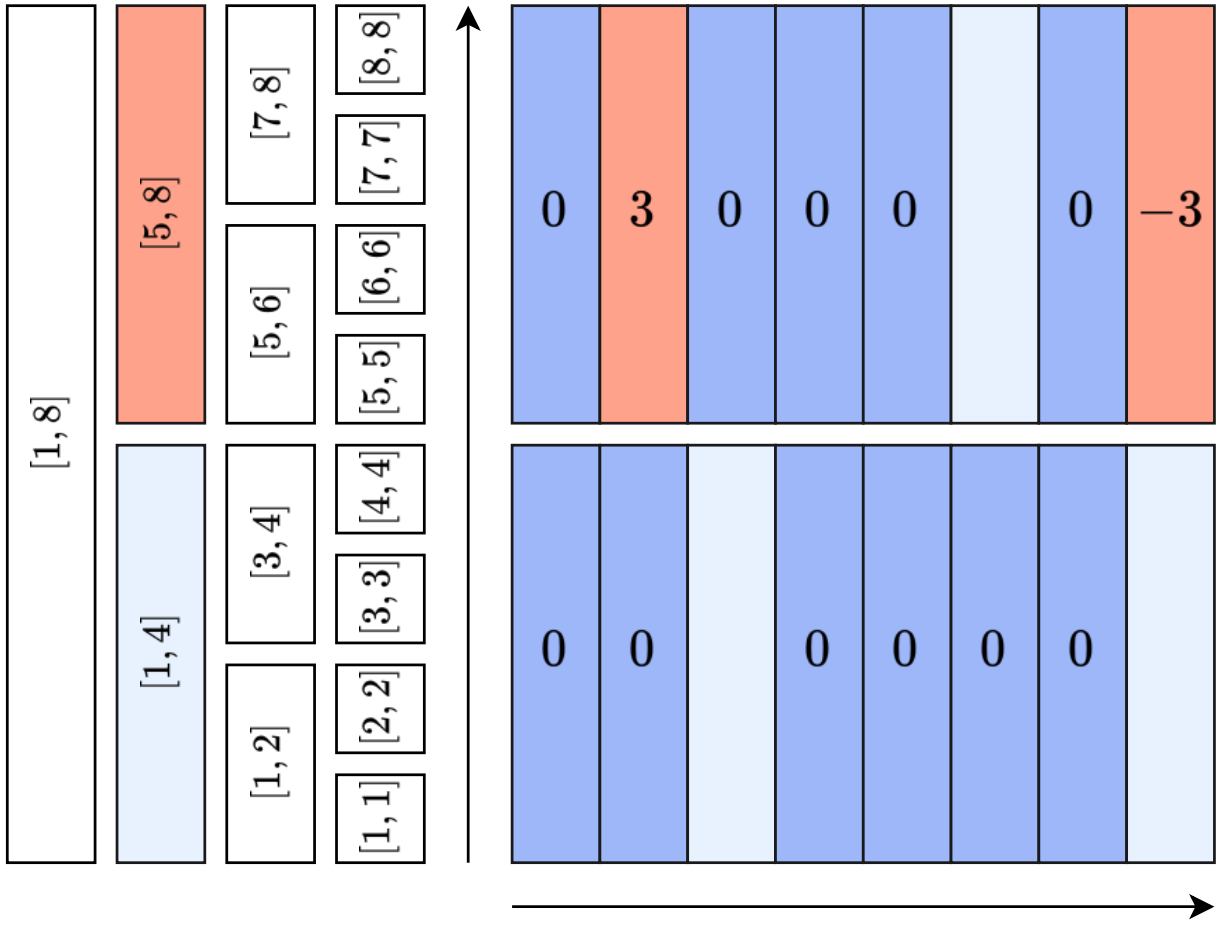
問題１２　倉庫管理ロボット

荷物追加クエリ(3 / 9)

1 2 7 6 3

- 箱 $2, 3, \cdots, 7$ に重さ 6 のボールを 3 個追加
- $(2, 6)$ に $+3$
- $(8, 6)$ に -3

根の方に移動しながら各ノードのセグメント木を更新していく



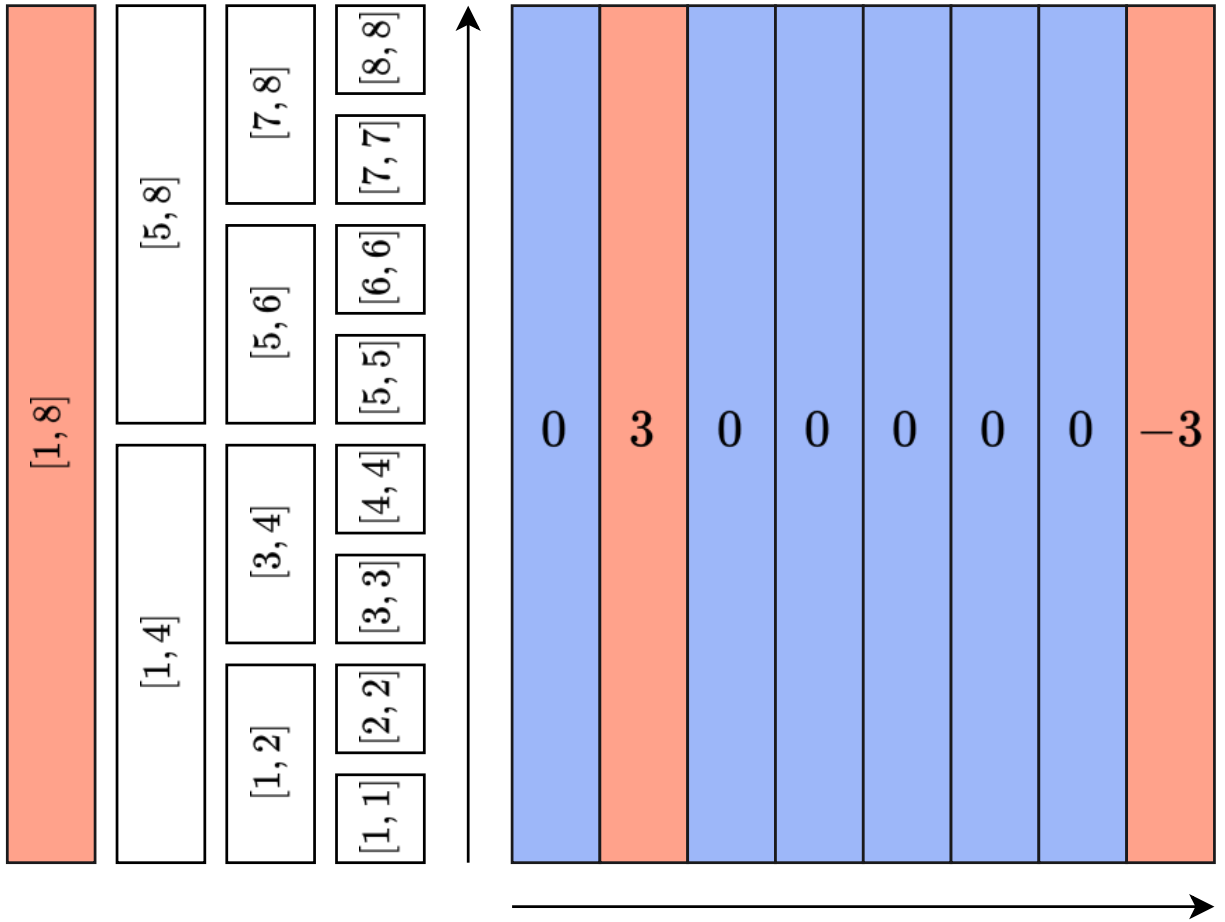
問題１２　倉庫管理ロボット

荷物追加クエリ(4/ 9)

1 2 7 6 3

- 箱 $2, 3, \cdots, 7$ に重さ 6 のボールを 3 個追加
- $(2, 6)$ に $+3$
- $(8, 6)$ に -3

根の方に移動しながら各ノードのセグメント木を更新していく



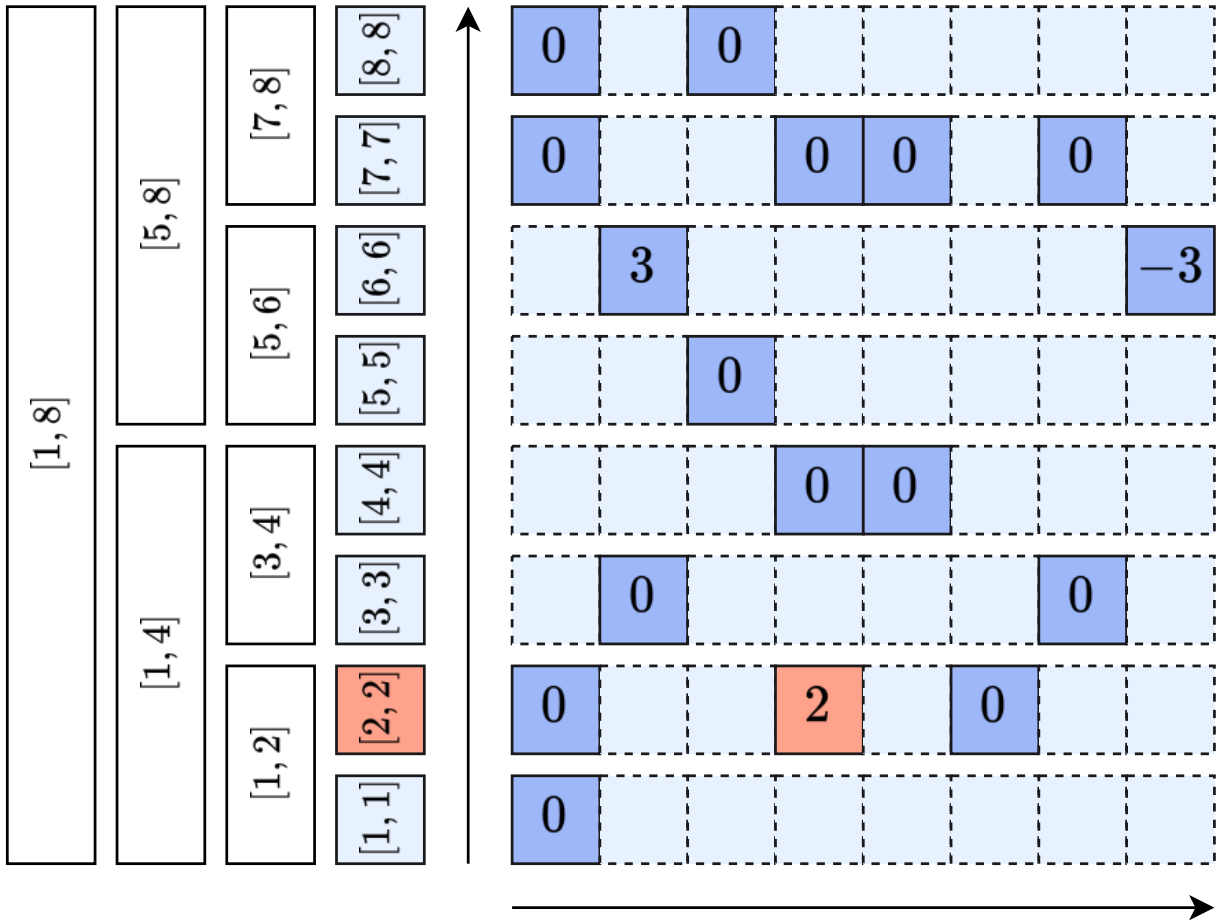
問題１２　倉庫管理ロボット

荷物追加クエリ(5 / 9)

1 4 8 2 2

- 箱 $4, 5, \cdots, 8$ に重さ 2 のボールを 2 個追加
- $(4, 2)$ に $+2$
- $(9, 2)$ に -2

$N < 9$ なので $(9, 2)$ への -2 の加算はする必要ない



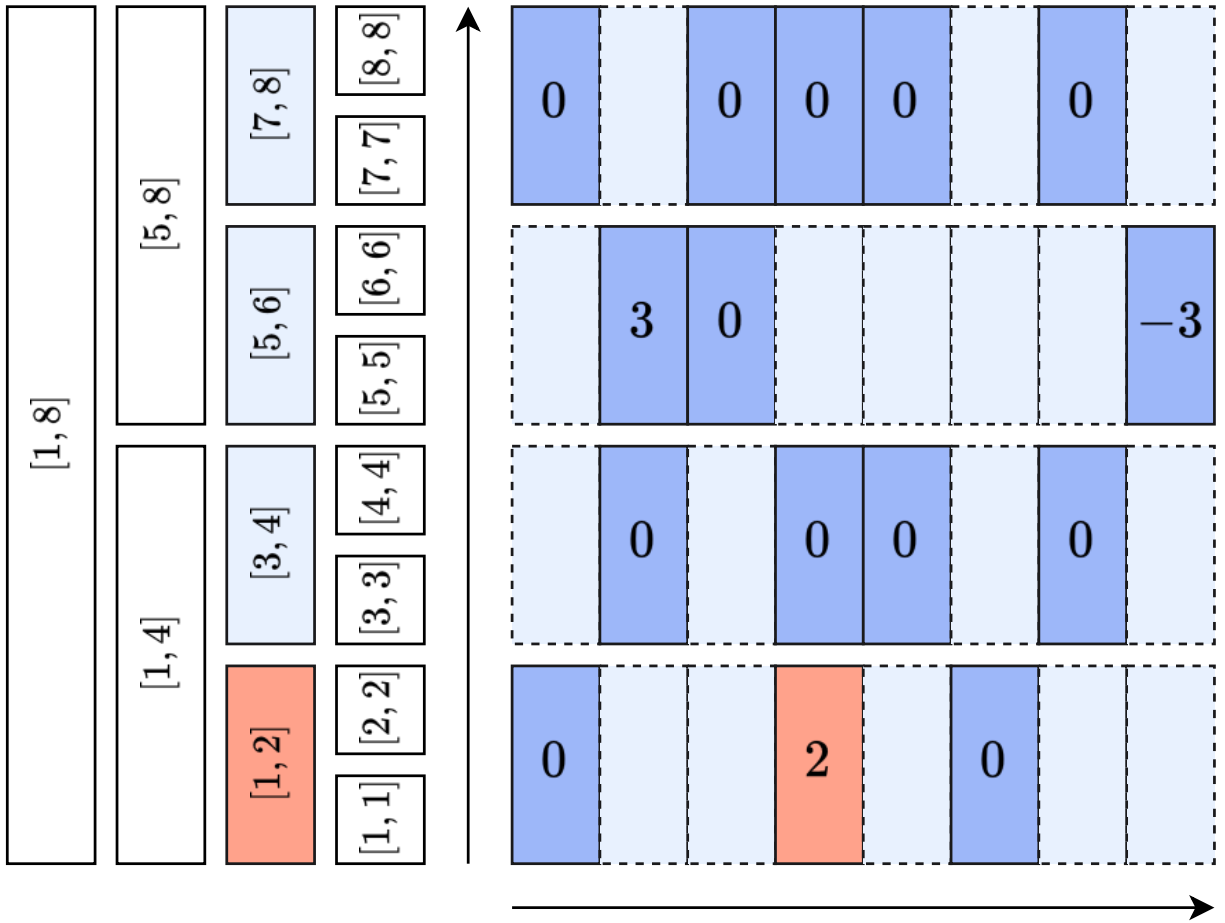
問題１２　倉庫管理ロボット

荷物追加クエリ(6 / 9)

1 4 8 2 2

- 箱 $4, 5, \cdots, 8$ に重さ 2 のボールを 2 個追加
- $(4, 2)$ に $+2$
- $(9, 2)$ に -2

$N < 9$ なので $(9, 2)$ への -2 の加算はする必要ない



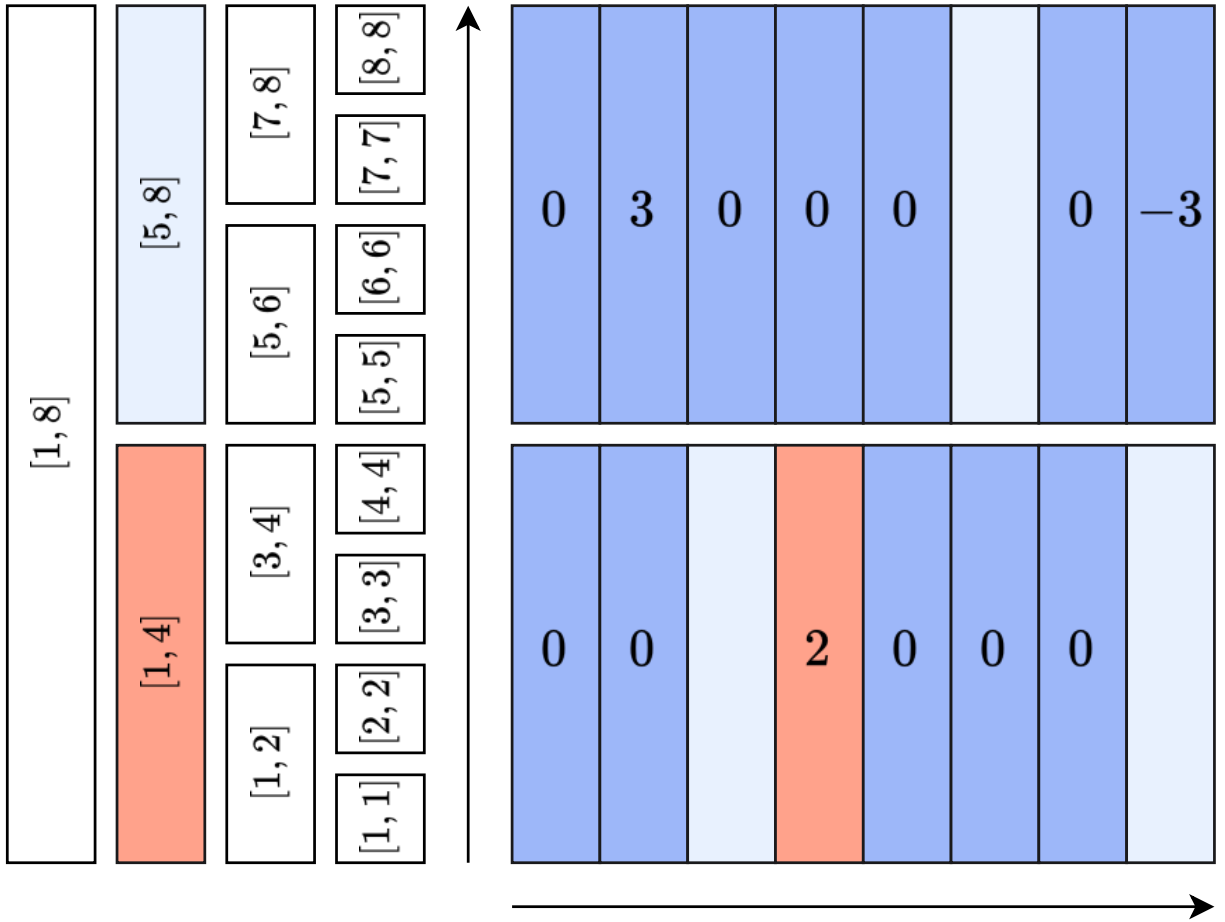
問題１２　倉庫管理ロボット

荷物追加クエリ(7 / 9)

1 4 8 2 2

- 箱 $4, 5, \cdots, 8$ に重さ 2 のボールを 2 個追加
- $(4, 2)$ に $+2$
- $(9, 2)$ に -2

$N < 9$ なので $(9, 2)$ への -2 の加算はする必要ない



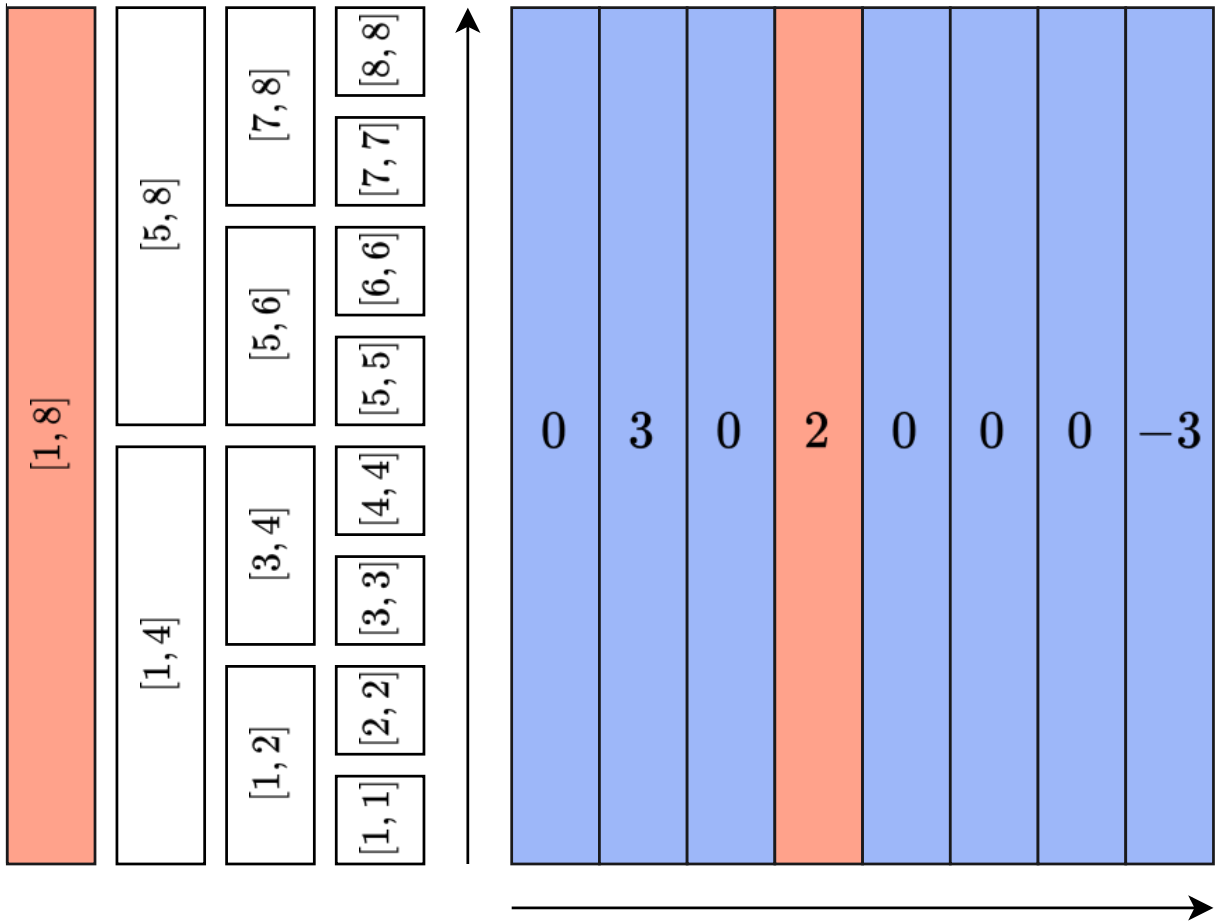
問題１２　倉庫管理ロボット

荷物追加クエリ(8 / 9)

1 4 8 2 2

- 箱 $4, 5, \cdots, 8$ に重さ 2 のボールを 2 個追加
- $(4, 2)$ に $+2$
- $(9, 2)$ に -2

$N < 9$ なので $(9, 2)$ への -2 の加算はする必要ない

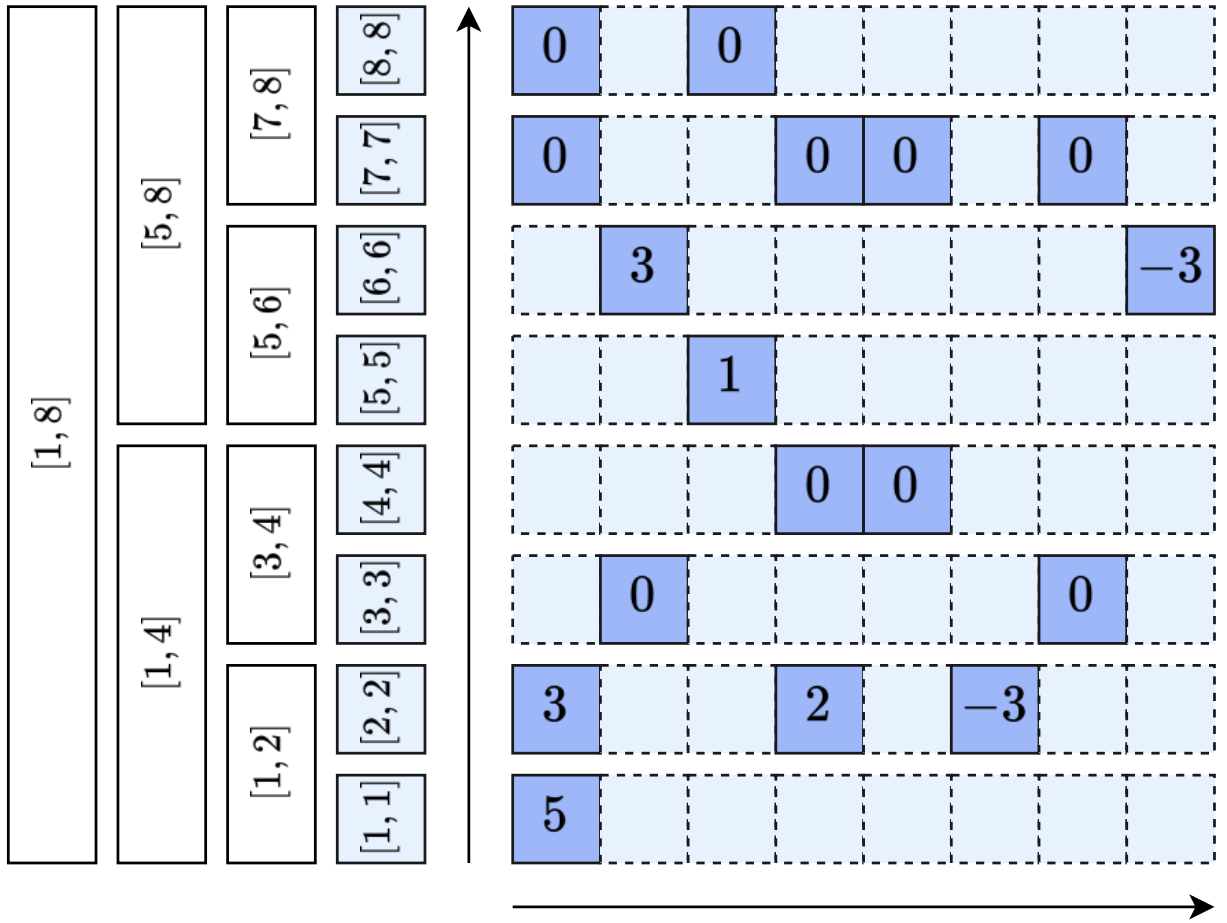


問題１２　倉庫管理ロボット

荷物追加クエリ(9 / 9)

以下の三つは省略

- 1 1 8 1 5
- 1 3 8 5 1
- 1 1 5 2 3

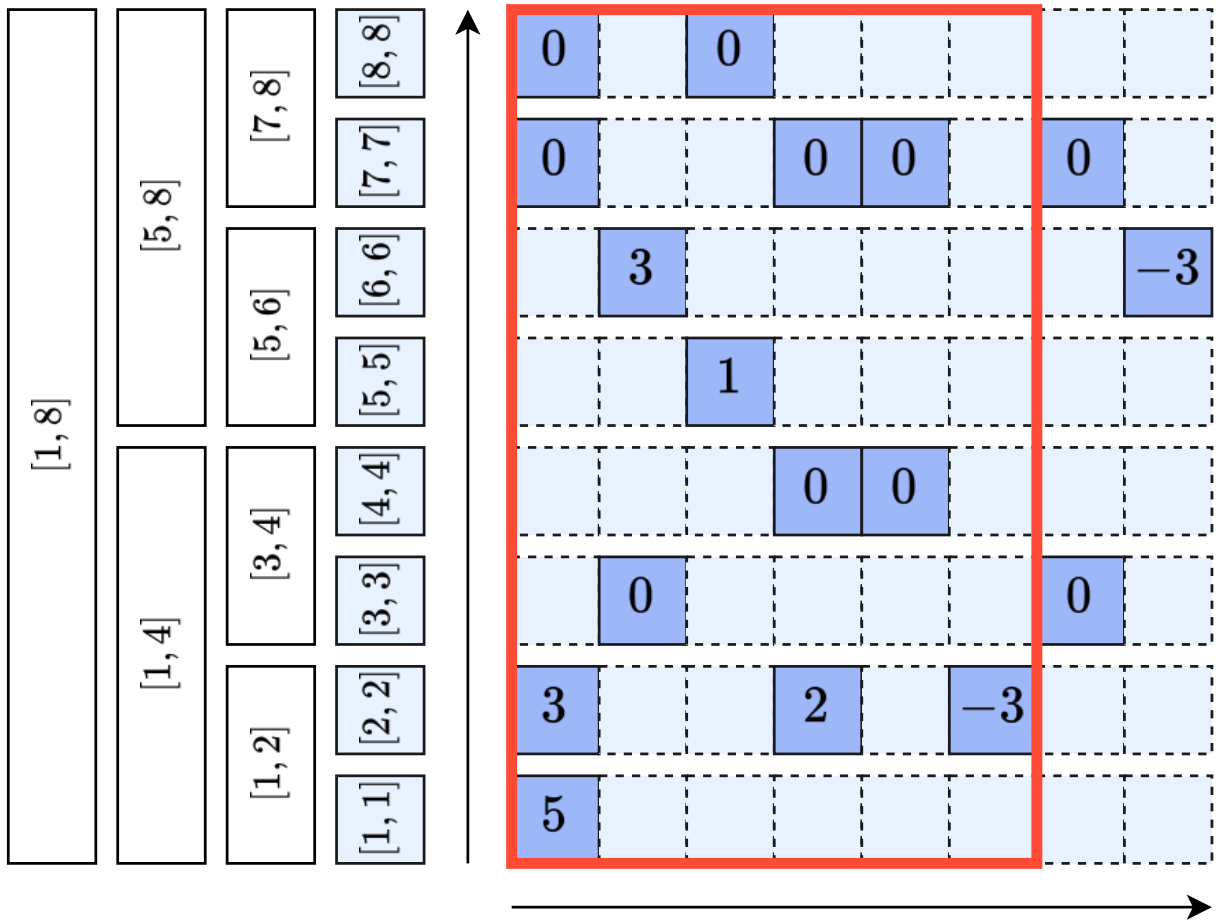


問題１２　倉庫管理ロボット

箱に含まれる荷物

例えば、右図の赤枠内の総和は箱 6 に含まれる荷物の個数と等しい

- $1:5$ 個
- $2:3+2-3=2$ 個
- $5:1$ 個
- $6:3$ 個



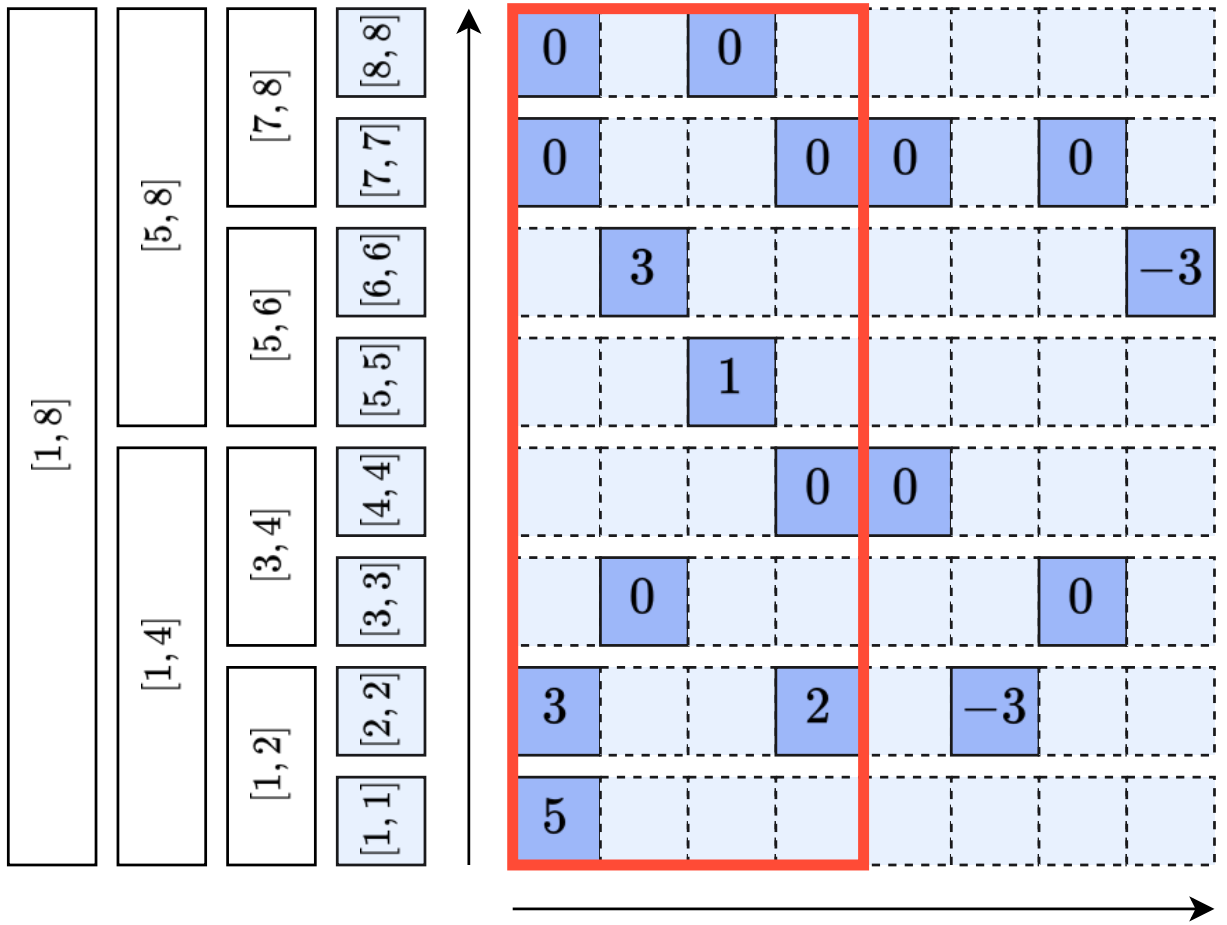
問題１２　倉庫管理ロボット

求値クエリ(1 / 5)

2 4 7

- 箱 4 に含まれる軽い方から 7 番目のボールの重さを求める

セグメント木上で二分探索をして, 重さ v 以下の荷物の個数が k 個以上となるような最小の v を求める



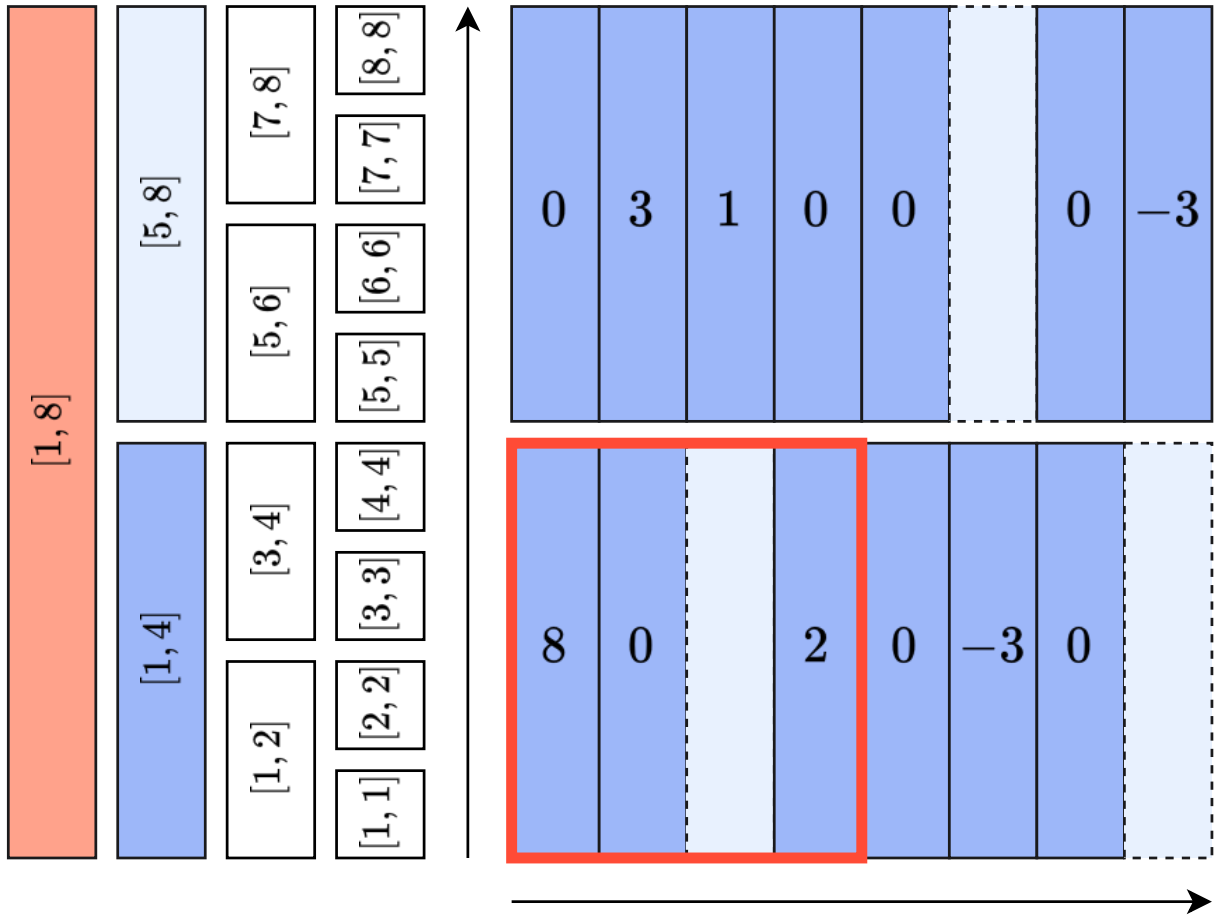
問題１２　倉庫管理ロボット

求値クエリ(2 / 5)

$[1,4] \times [1,4]$ の総和は 10

- 箱 4 に含まれる重さ 1 以上 4 以下の荷物の個数は 10 個 $(\geq k = 7)$

7 番目のボールの重さは 1 以上 4 以下になるので、左の子ノードに移動する



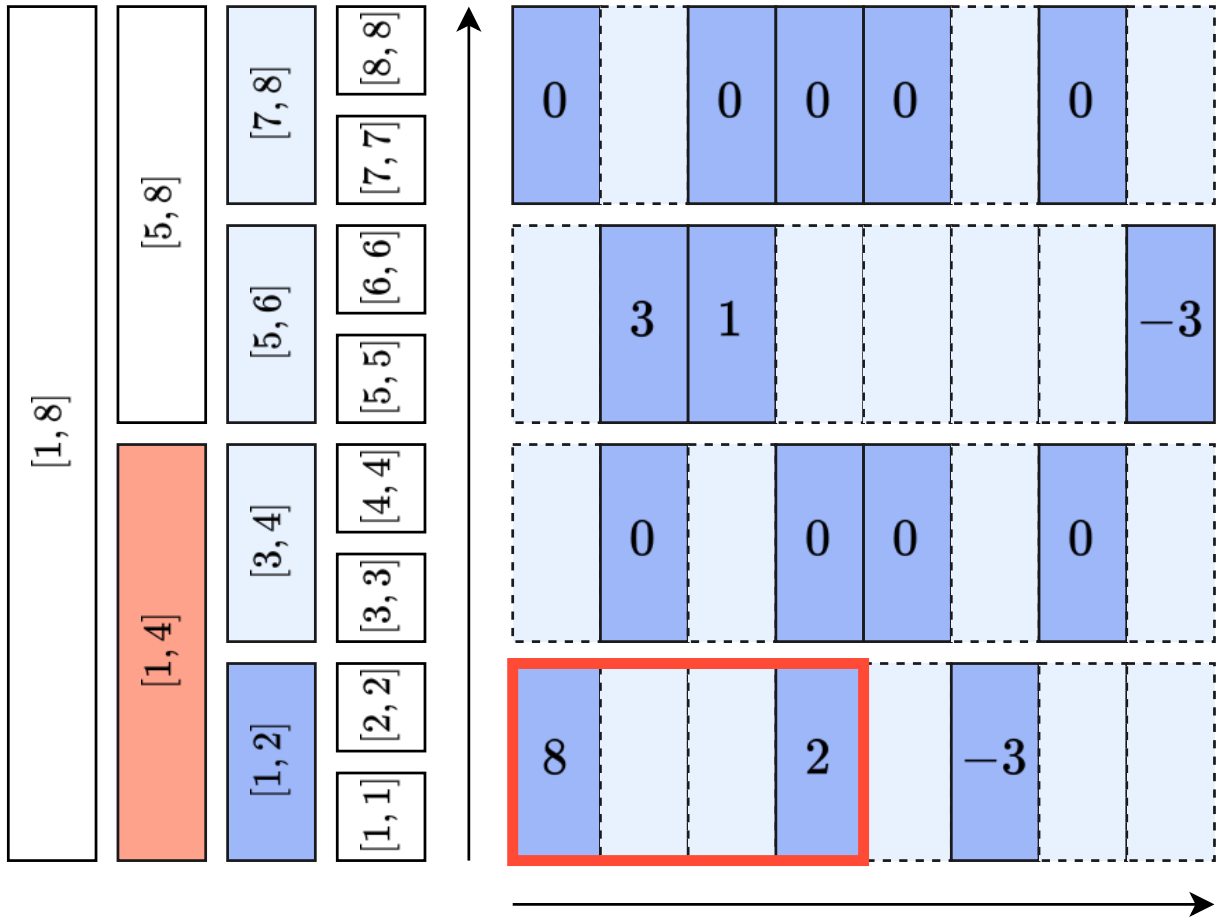
問題１２　倉庫管理ロボット

求値クエリ(3 / 5)

$[1,4] \times [1,2]$ の総和は 10

- 箱 4 に含まれる重さ 1 以上 2 以下の荷物の個数は 10 個 $(\geq k = 7)$

7 番目のボールの重さは 1 以上 2 以下になるので、左の子ノードに移動する



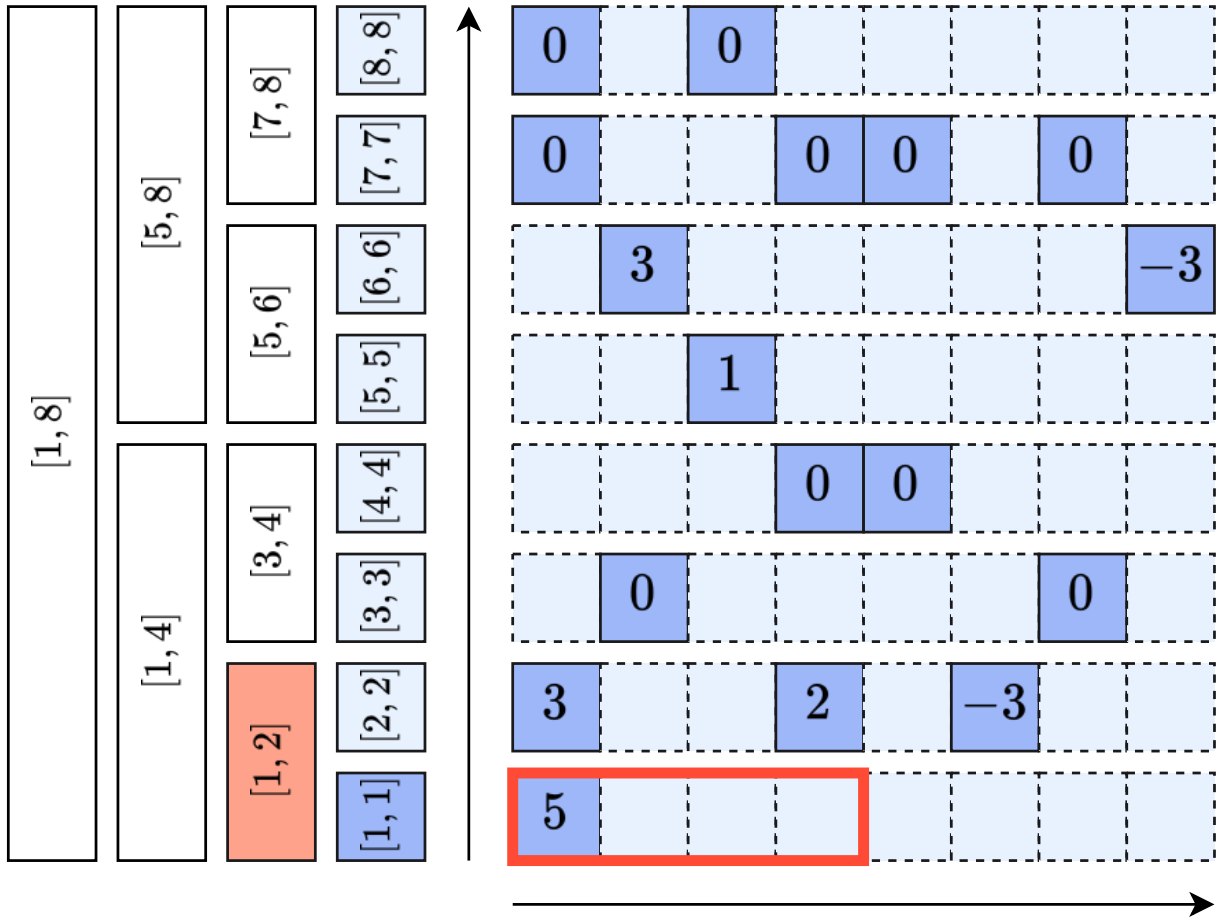
問題１２　倉庫管理ロボット

求値クエリ(4 / 5)

$[1,4] \times [1,1]$ の総和は 5

- 箱 4 に含まれる重さ 1 以上 1 以下の荷物の個数は 5 個 $(< k = 7)$

7 番目のボールの重さは 1 以上 1 以下にはならないので、$k \leftarrow 7 - 5 = 2$として右の子ノードに移動する

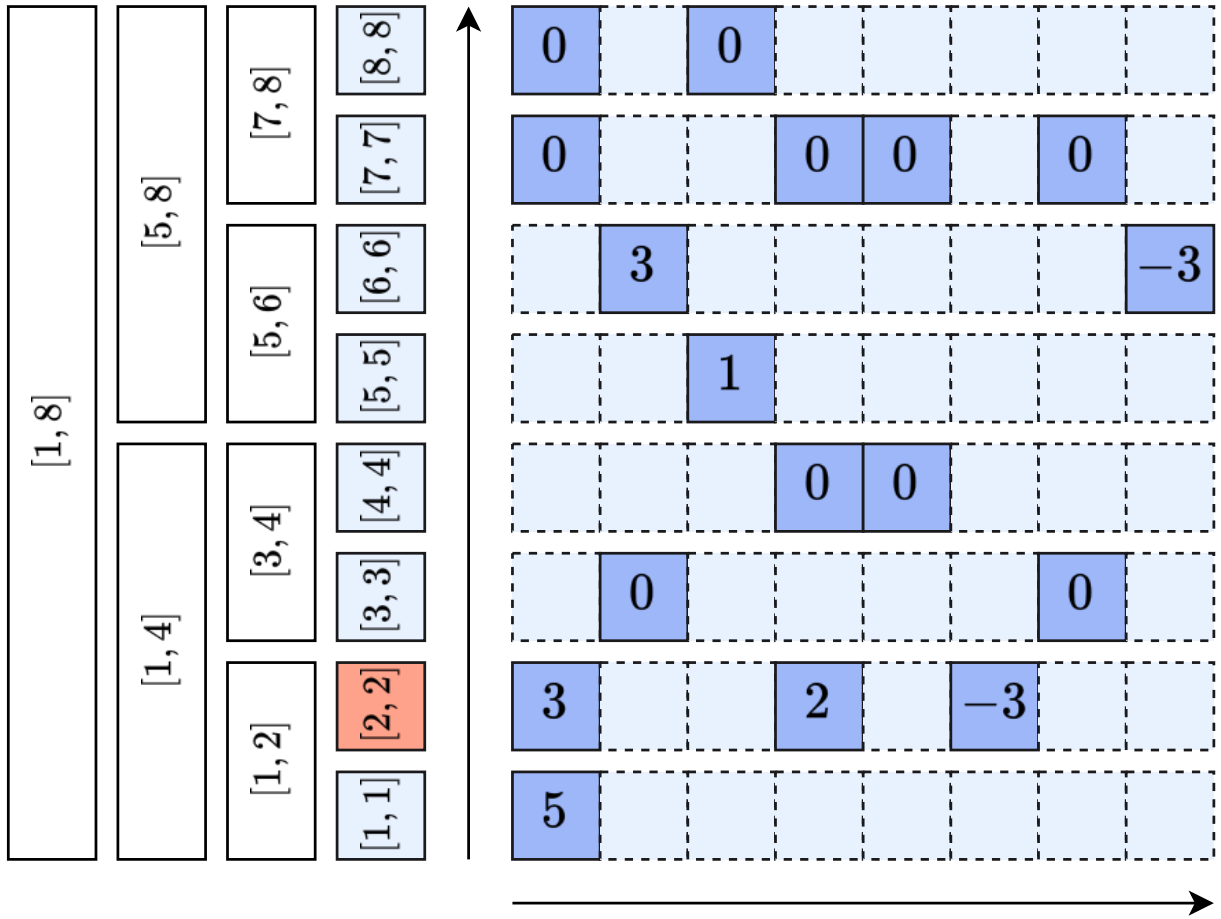


問題１２　倉庫管理ロボット

求値クエリ(5 / 5)

たどり着いた葉の値が答え

- 箱 4 に含まれる軽い方から 7 番目のボールの重さは 2



問題１２　倉庫管理ロボット

計算量

荷物追加クエリ一回あたり

- $O(\log N \log Q)$

求値クエリ一回あたり

- $O(\log N \log Q)$

全体

- $O(Q \log N \log Q)$

問題１２　倉庫管理ロボット

その他の解法

セグメント木 + binary trie

- 各ノードにbinary trieを乗せて多重集合を管理
- 荷物追加クエリはいくつかの二冪長の区間に分割
- 求値クエリは $\log N$ 個のtrieを並べて二分探索

セグメント木 + 並列二分探索

$\sqrt{Q}$ 分木 + Mo's algorithm

- 二次元セグメント木と同様のことができる

ご視聴ありがとうございました。
また来年2025で!!