

# しゃくとり法

Keiya Sato  
Yokoyama Lab, M1  
May 02, 2023

# 目次

1. しゃくとり法とは
2. 問題
  1. 方針
  2. 解法①
  3. 解法②
3. 補足: 区間の数え上げ
4. まとめ
5. 例題: ABC032 C - 列

# しゃくとり法とは (1/2)

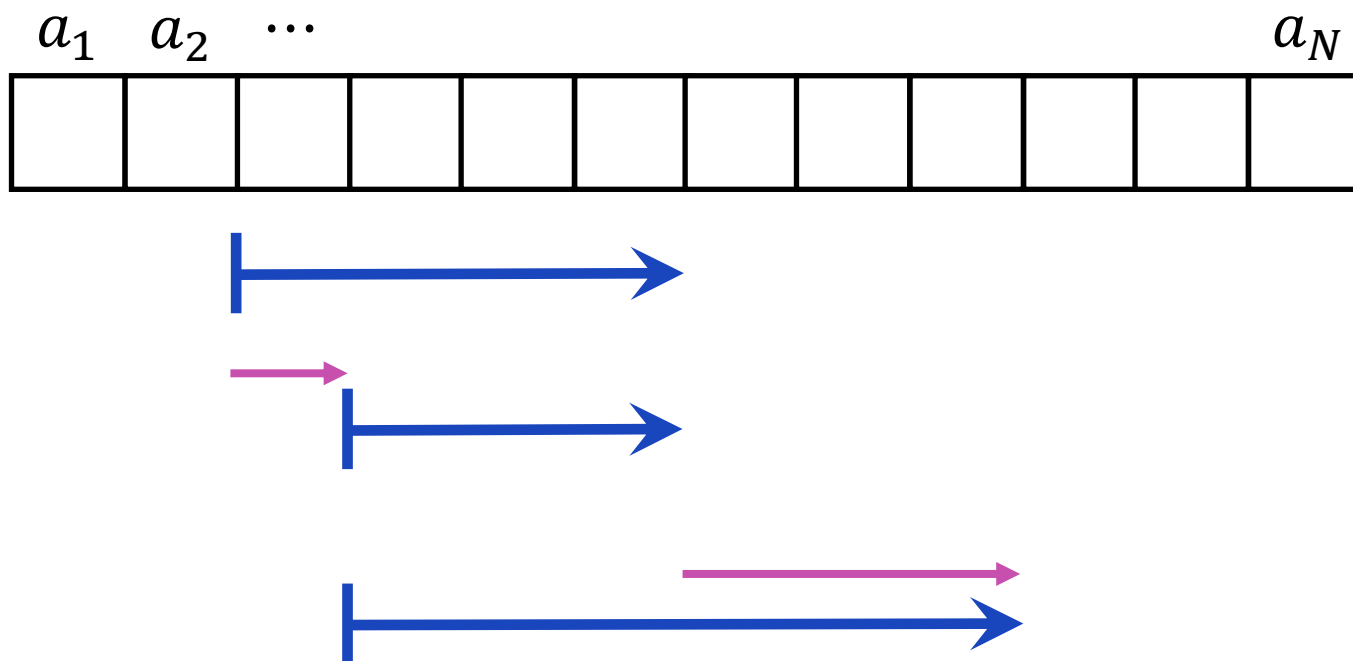
以下のような問題を解く際に使えるアルゴリズム

- 条件を満たす区間のうち最大のものを求める
- 条件を満たす区間のうち最小のものを求める
- 条件を満たす区間を全て数え上げる

**「数列に対し条件を満たす区間を求める」 ことができる**

# しゃくとり法とは (2/2)

- 英語では “Two-Pointers”, “Sliding Window” と呼ばれる
  - 区間の左端と右端をずらしながら問題を解く



# 問題

## 問題文

長さ  $N$  の数列  $a_1, a_2, \dots, a_N$  と整数  $X$  が与えられます.  
 $a_l + a_{l+1} + \dots + a_{r-1} + a_r < X$  ( $1 \leq l \leq r \leq N$ ) を満たす最大の区間の長さを求めてください.

## 制約

- $1 \leq N \leq 10^5$
- $1 \leq a_i \leq 10^9$  ( $1 \leq i \leq N$ )
- $1 \leq X \leq 10^{14}$

(AOJ-DSL\_3\_C The Number of Windows 改題)

# 方針

求めるべきは何か？

区間和が  $X$  を超えないような最大の区間

## 【方針】

◆愚直に全区間調べるのは難しい

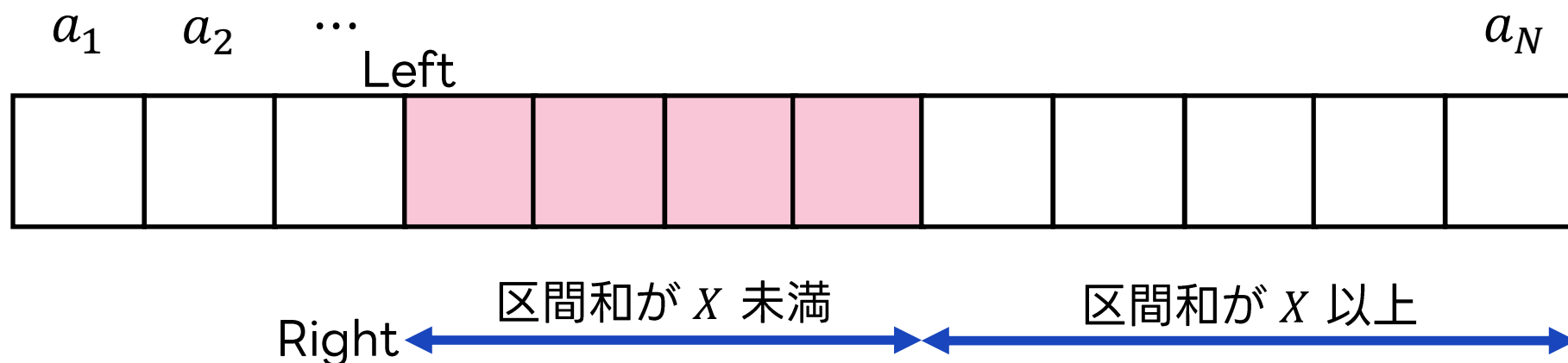
◆時間計算量は  $O(N^3)$

◆区間の左端を決めた時に右端が高速に求められるとうれしい

◆右端が  $O(X)$  で求められる時, 全体で  $O(NX)$

# [解法①] 累積和 + 二分探索

- 左端を固定した時, ある範囲までは区間和が  $X$  以下, ある範囲を超えると  $X$  以上になる
- 右端の位置で二分探索
  - 条件: 区間和が  $X$  を超えるかどうか



# [解法①] 時間計算量と高速化

- 左端を全て試す

→  $O(N)$

- 二分探索

→  $O(\log(N))$

- 区間和の計算

- 累積和が利用可能

→ 前処理:  $O(N)$

→ 区間和取得:  $O(1)$

➡ 全体で  $O(N\log(N))$

もっと高速化できないか？

【考察】

- ✓ 左端を決めた時に求められる区間は  $X$  を超えないような極大の区間
- ✓ 左端をずらしたとき、右端は必ず前回より右に移動する
- ✓ 毎回二分探索するのではなく、右端も右にずらしながら走査できないか？



# **【解放②】しゃくとり法 (1/2)**

区間和を変数に保持しながら以下を繰り返す

1. 左端を決めた時, 条件を満たす間右端を移動させる
2. 区間が極大の時, 左端を移動させる

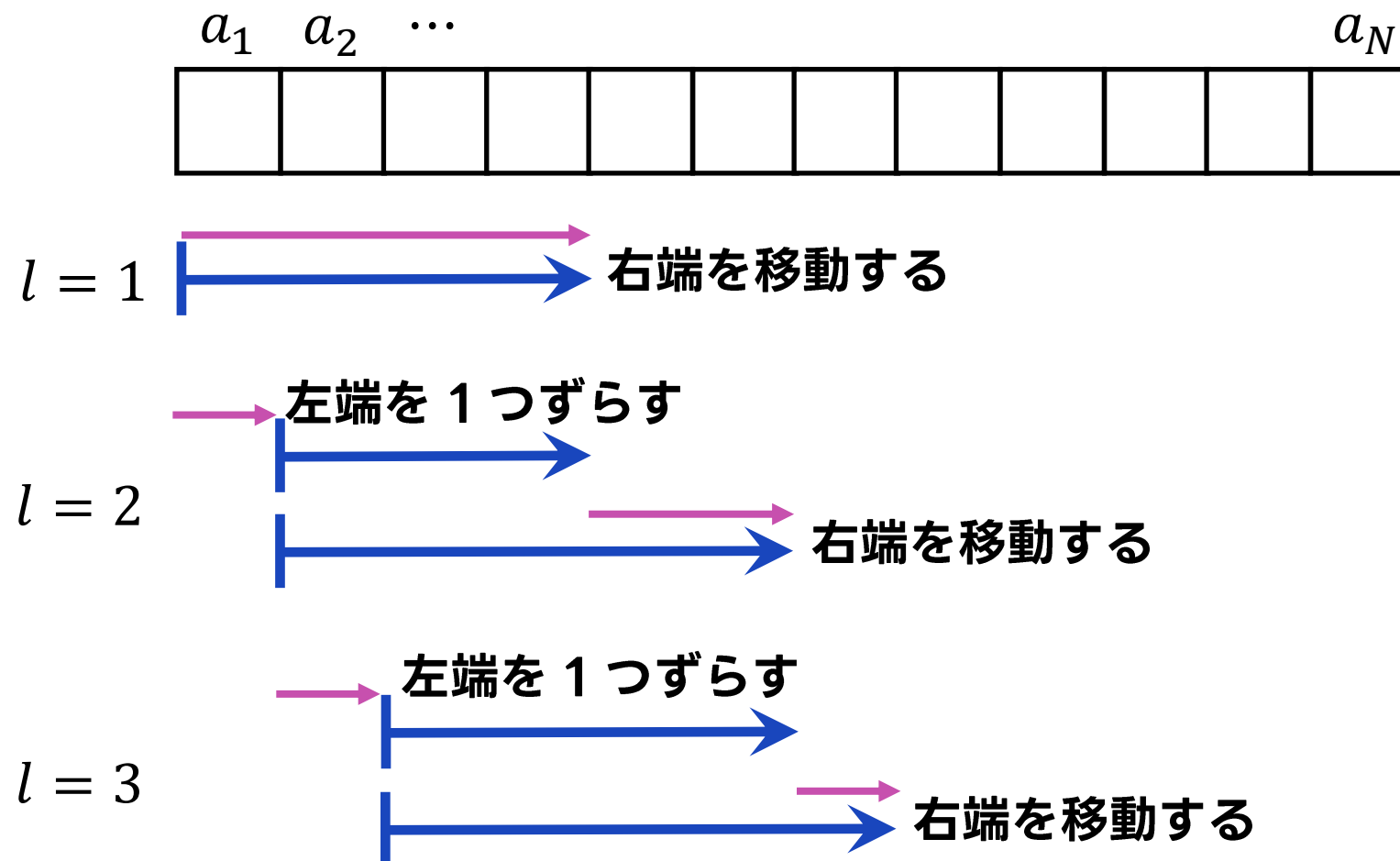
- 右端の移動

- 区間和に  $a_r$  を加算する

- 左端の移動

- 区間和から  $a_l$  を減算する

# [解放②] しゃくとり法 (2/2)



# [解放②] 時間計算量

- 左端は数列の各要素を 1 度だけ見る  
→  $O(N)$
- 右端も数列の各要素を 1 度だけ見る  
→  $O(N)$
- 区間や区間和, 答えの更新  
→  $O(1)$

➡ 全体で  $O(N)$



# [解放②] 実装

## C++

```
int r = 0;
int s = 0;
int ans = 0;
for (int l = 0; l < n; l++) {
    while (r < n and s < x) {
        s += a[r];
        r += 1;
    }
    ans = max(ans, r - l);
    s -= a[l];
}
```

## Python

```
r = 0
s = 0
ans = 0
for l in range(n):
    while r < n and s < x:
        s += a[r]
        r += 1
    ans = max(ans, r - l)
    s -= a[l]
```

# しゃくとり法まとめ

- 条件を満たす区間を計算するような問題で利用できる可能性がある
  - 条件を満たす最大・最小の区間
  - 条件を満たす区間の数
- 多くの場合二分探索で代替可能
  - ただし時間計算量で劣る
- 左端と右端を少しずつ移動させながら条件を満たす区間を列挙する
  - 時間計算量:  $O(N)$

# 例題1: ABC032 C - 列

- 非負整数列の中で区間積が  $K$  以下であるような最長の区間の長さを求める
- 理論上は二分探索でも解ける
  - 実際は累積積が大きすぎて扱うことが難しい
- 区間の端を移動するときの操作は以下の通り
  - 右端: 乗算
  - 左端: 除算

## 例題2: ABC038 C – 単調増加

- 数列のうち単調増加になっている区間の数を求める
- ヒント
  - ある区間が単調増加のとき, そこに含まれるあらゆる区間も単調増加になる

## 例題3: ABC229 D - Longest X

- $X$  と  $.$  からなる文字列  $S$  に対し「 $.$  を  $X$  に置き換える」という操作を行う
- $K$  回以下の操作で  $X$  を最大何個連続させることができるか
- ヒント
  - 「 $S$  の中で  $.$  が  $K$  個以下である最長の区間の長さはいくらか」という問題に言い換えることができる
  - 上記の区間はしゃくとり法で計算可能