

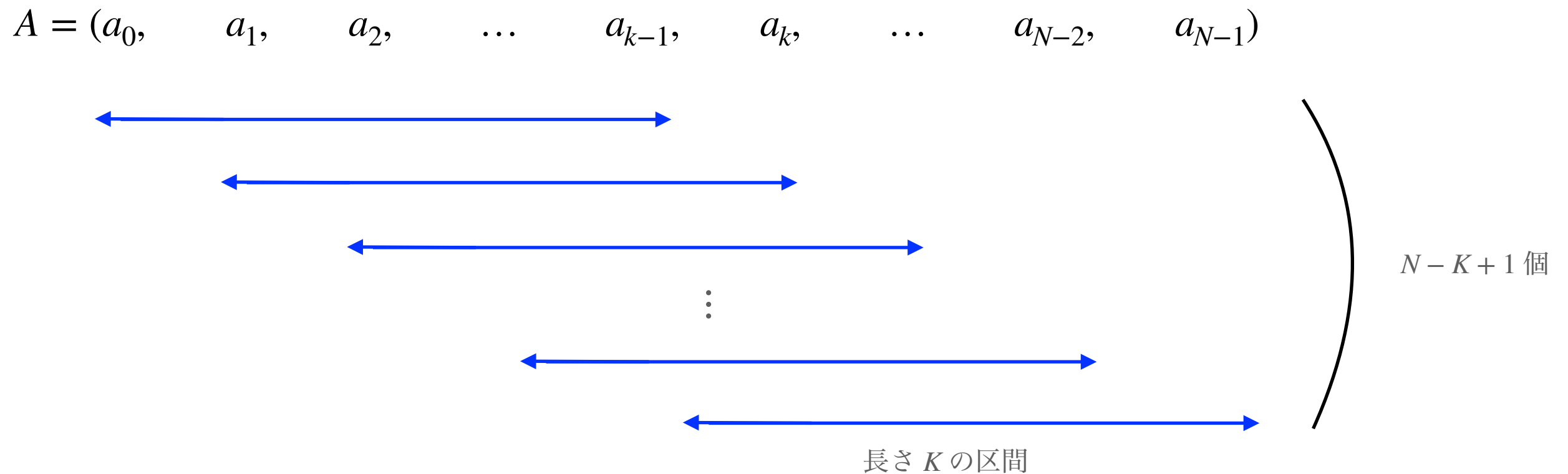
# 6 問目

## ABC037 - C 総和

### 解説

# 問題概要

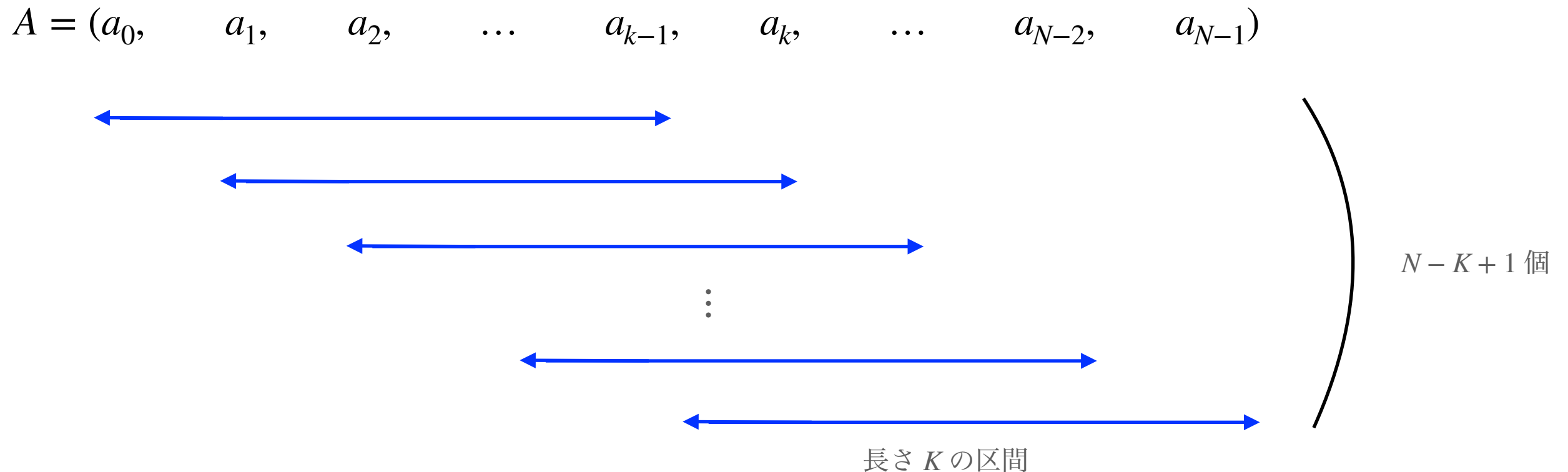
長さ  $N$  の数列  $A = (a_0, a_1, a_2, \dots, a_{N-2}, a_{N-1})$  と 1 以上  $N$  以下の整数  $K$  が与えられる。  
数列  $A$  には長さ  $K$  の連続する部分列が  $N - K + 1$  個あります。  
それぞれ部分列に含まれる値の合計の総和を求めよ。



問題文で言われた通りに実装した場合の計算量は？

# 問題概要

問題文で言われた通りに実装した場合の計算量は？



長さ  $K$  の区間の和

→  $K - 1$  回の足し算が必要

×

$N - K + 1$  個の区間の和を計算

→ 全体で  $(K - 1) \times (N - K + 1) = O(N^2)$  の計算量 (  $N$  を固定した時  $K = \frac{N}{2}$  付近が最大 )

$N \leq 10^5$ , Time Limit 2 sec の制約では、 $O(N^2)$  のプログラムでは **TLE** してしまう

# 解法 1

$$s_{l,r} = \sum_{i=l}^r a_i \text{ と定義する}$$

賢く長さ  $K$  の連続した部分列の和を計算する

$$\begin{array}{ccccccccccc} a_i & + & a_{i+1} & + & \dots & + & a_{i+K-2} & + & a_{i+K-1} & & = & s_{i,i+K-1} \\ & & a_{i+1} & + & a_{i+2} & + & \dots & + & a_{i+K-1} & + & a_{i+K} & = & s_{i+1,i+K} \end{array}$$

$$s_{i+1,K} = s_{i,i+K-1} - a_i + a_{i+K}$$

↑ の漸化式を用いると、 $s_{i,i+K-1}$  から  $s_{i+1,K}$  を高速に計算できる

計算量は？

$s_{i,i+K-1}(i=0)$  の計算  $\rightarrow O(K)$  で素直に計算

$s_{i,i+K-1}(i>0)$  の計算  $\rightarrow$  漸化式を使い  $O(1)$  で計算

計算量は全体で  $O(N)$

このアルゴリズムを実装すれば AC できる

# 解法 1

賢く長さ  $K$  の連続した部分列の和を計算する

```
#include <iostream>
#include <vector>
using namespace std;

int main(){
    int n, k;
    cin >> n >> k;
    vector<long long> A(n);
    for(int i = 0; i < n; ++i)
        cin >> A[i];

    long long s = 0;
    for(int i = 0; i < k; ++i)
        s += A[i];

    long long ans = s;
    for(int i = k; i < n; ++i){
        s += A[i] - A[i-k];
        ans += s;
    }

    cout << ans << endl;
}
```

← 入力受け取り

←  $s_{0,K-1}$  を計算

←  $s_{i,i+K-1} (i > 0)$  の計算

# 解法 2

## 累積和

$$A = (a_0, a_1, a_2, \dots, a_{k-1}, a_k, \dots, a_{N-2}, a_{N-1})$$

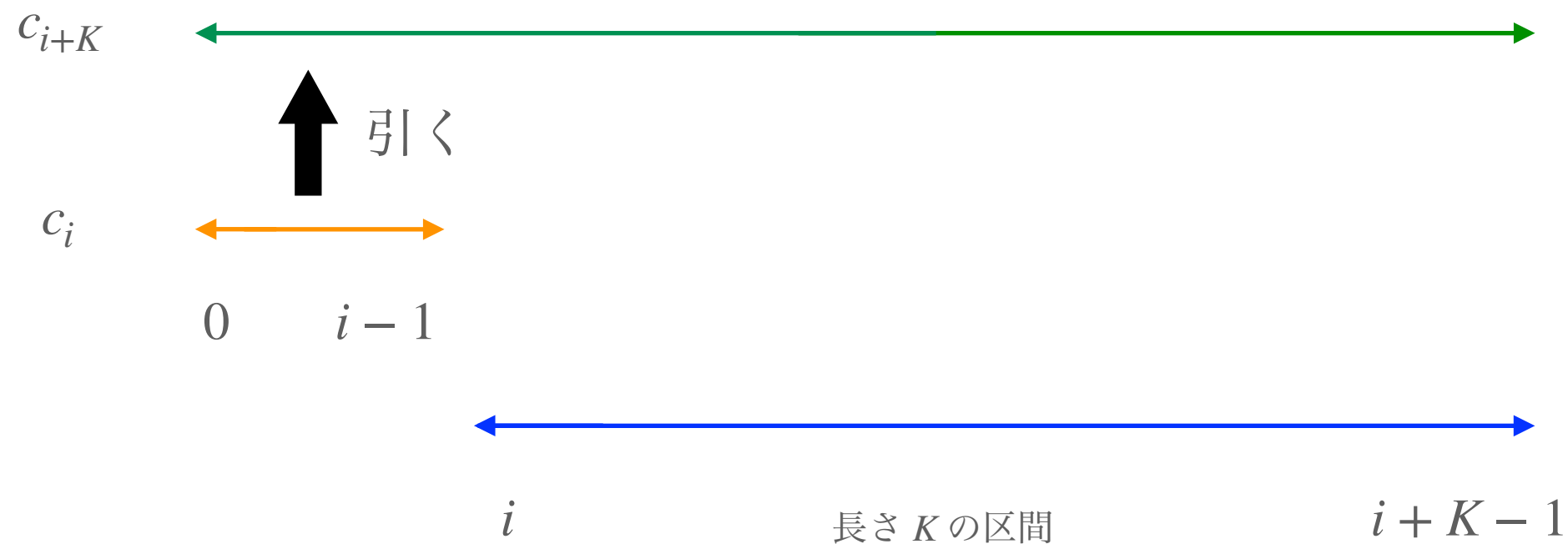


$$c_i = \sum_{k=0}^{i-1} a_k \text{ で定義される配列を累積和と呼ぶ}$$

累積和を使うと、長さ  $K$  の区間の和を  $O(1)$  の計算量で計算できる！

# 解法 2

## 累積和



$$s_{i,i+K-1} = c_{i+K} - c_{i-1}$$

累積和を使うと、長さ  $K$  の区間の和を  $O(1)$  の計算量で計算できる！

→ 全体で  $O(N)$  の計算量 → AC

# 解法 2

## 累積和

```
#include <iostream>
#include <vector>
using namespace std;

int main(){
    int n, k;
    cin >> n >> k;
    vector<long long> A(n);
    for(int i = 0; i < n; ++i)
        cin >> A[i];

    vector<long long> C(n+1);
    for(int i = 1; i <= n; ++i)
        C[i] += A[i-1] + C[i-1];

    long long ans = 0;
    for(int i = k; i <= n; ++i){
        ans += C[i] - C[i-k];
    }
    cout << ans << endl;
}
```

← 入力受け取り

← 累積和を計算

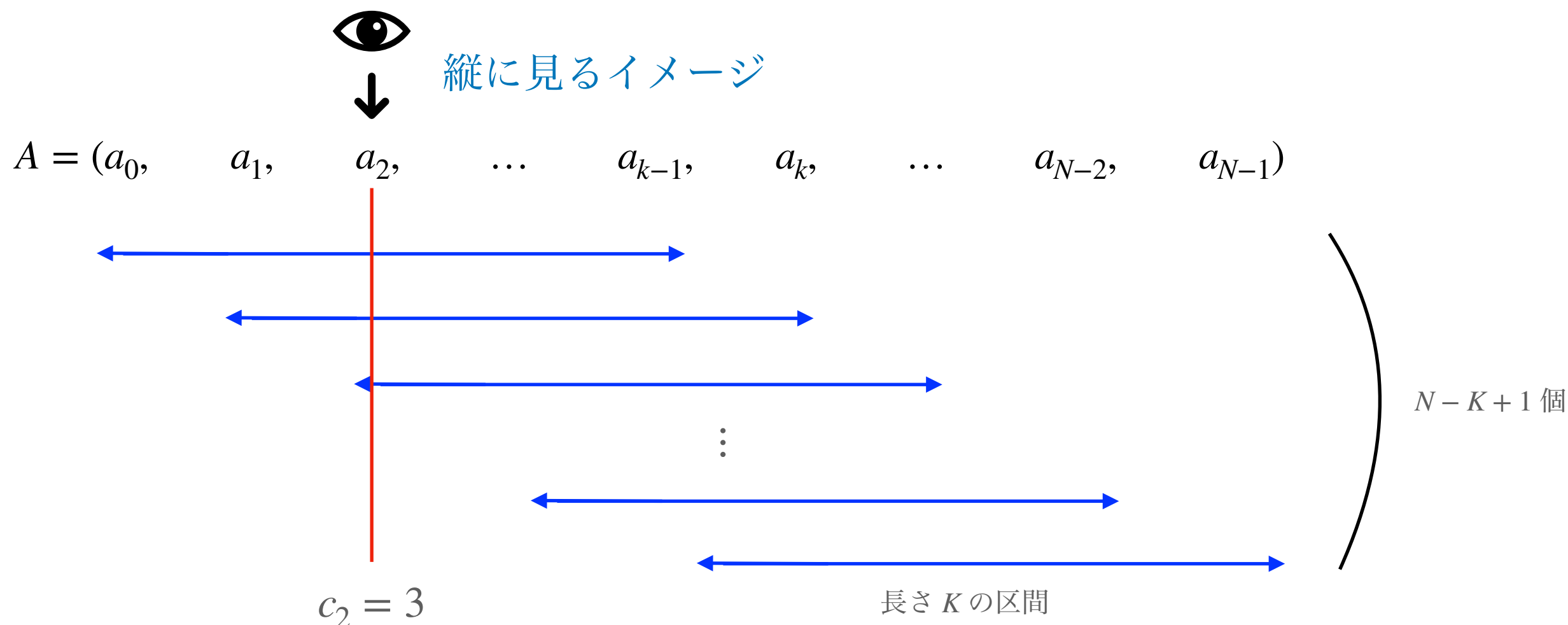
←  $s_{i,i+K-1} (i > 0)$  の計算



# 解法 3

## 各要素の寄与を考える

答えは  $\sum_{i=0}^{N-1} c_i a_i$  という形になっている。  $c_i$  が分かれば答えを計算できる。



$c_i = \min(N - K, i) - \max(0, i - K + 1) + 1$  という式で表せます (考えてみてください)

# 解法 3

## 各要素の寄与を考える

```
#include <iostream>
#include <vector>
using namespace std;

int main(){
    int n, k;
    cin >> n >> k;
    vector<long long> A(n);
    for(int i = 0; i < n; ++i)
        cin >> A[i];

    long long ans = 0;
    for(int i = 0; i < n; ++i){
        long long c = min(n-k, i) - max(0, i-k+1) + 1;
        ans += A[i]*c;
    }
    cout << ans << endl;
}
```