


# レギオ

## 初級編

# 演習の目標

- JOIの予選を突破するために必要な**アルゴリズム**の知識を身につける
- JOIの本選で戦うために必要な**アルゴリズム**の知識を身につける
- AtCoderで緑になるために必要な**アルゴリズム**の知識を身につける

AtCoderの上位30%！

(<https://chokudai.hatenablog.com/entry/2019/02/11/155904>)

# 予選突破/本選/緑に必要なアルゴリズム

- 全探索
  - 順列全探索
  - bit全探索
  - 深さ優先探索
  - 幅優先探索
- 二分探索
- 動的計画法(DP)
  - bitDP
  - 確率DP
- プライオリティキュー
- Union-Find
- imos法
- セグメント木
- 木の直径
- 最短路問題
  - ダイクストラ法
  - ワーシャルフロイド法
- 素数/約数
  - エラトステネスの篩
  - 高速な素数判定
  - 高速な約数列挙
  - 高速な素因数分解
- modの性質
- 繰り返し二乗法

# 今日の目標

- ・ プログラマから、**競技プログラマ**になる
  - ・ **速度**を意識してプログラムを書く
  - ・ **アルゴリズム・データ構造**の知識を身につける

# 概要

- ・ 計算量

- ・ コードの計算量を答えられるようになるう
- ・ コードを書く前に計算量を見積もれるようになるう

- ・ アルゴリズム

- ・ 二分探索を理解し，使いこなせるようになるう
- ・ imos法を理解し，使いこなせるようになるう

# 競技プログラミングって何？

## 問題

AとBが与えられる.  $A+B$ を求めよ.

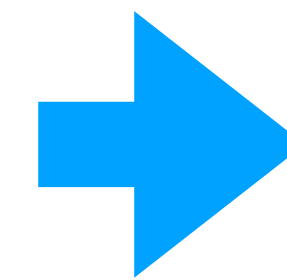
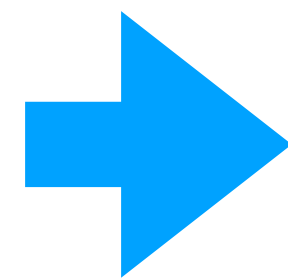
## プログラム

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int a, b;
    cin >> a >> b;
    cout << a + b << endl;
}
```

入力

13 24



出力

37

↑  
ここを書こう！

# 競技プログラミングは楽しい！

問題を読む → どんなアルゴリズムで解けるのかを考える → コードを書く

↑  
ここが楽しい！

正しく

高速に

# AtCoderに登録しよう

- AtCoder

<https://atcoder.jp/>

- スプレッドシート

<https://docs.google.com/spreadsheets/d/1jlvopGcBqWbt8hkXJ-6wTYgJgfMXPAPbr2tbGd0qoNE/edit?usp=sharing>



# 実行環境

- 自分のパソコンにC++の実行環境がすでにある  
→自分のパソコンで実行して大丈夫です
- C++の実行環境がない  
→AtCoderのコードテストを使いましょう  
[https://atcoder.jp/contests/practice/custom\\_test](https://atcoder.jp/contests/practice/custom_test)

# コードテストの使い方

1. 言語はC++ (GCC 9.2.1)を選ぶ

言語

C++ (GCC 9.2.1)

2. プログラムを書く

3. 入力を与える

4. 実行ボタンを押す

5. 出力, エラーを確認める

コードテスト

言語

C++ (GCC 9.2.1)

ソースコード

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int a, b;
6     cin >> a >> b;
7     cout << (a + b) << endl;
8 }
9
10
```

📁 ファイルを開く

エディタ切り替え

高さ自動調節

※512 KiB まで

※ソースコードは「Main.拡張子」で保存されます

標準入力

5 3

※512 KiB まで

※AtCoderの入力データでは通常、末尾にも改行が入ります

実行

終了コード

0

実行時間

6 ms

メモリ

424 KB

標準出力

8

標準エラー出力

# 解いてみよう！

- ・ アカウントを作ってログイン

<https://atcoder.jp>

右上の新規登録からアカウントを作成/ログイン

- ・ **Welcome to AtCoder**を解いてみよう！

[https://atcoder.jp/contests/abs/tasks/practice\\_1](https://atcoder.jp/contests/abs/tasks/practice_1)

言語は**C++ (GCC 9.2.1)**を選んでください

→ 言語

C++ (GCC 9.2.1)

結果

AC

**AC**になれば正解！

# 回答例

## 問題文

高橋君はデータの加工が行いたいです。

整数  $a, b, c$  と、文字列  $s$  が与えられます。  $a + b + c$  の計算結果と、文字列  $s$  を並べて表示しなさい。

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    int a, b, c;
    string s;
    cin >> a >> b >> c >> s;
    cout << (a+b+c) << " " << s << endl;
}
```

← 問題を正しく理解

## 入力

入力は以下の形式で与えられる。

```
a
b c
s
```

## 出力

$a + b + c$  と  $s$  を空白区切りで1行に出力せよ。

← 指定された形式で入力

← 指定された形式で出力

計算量を求めて  
実行時間を見積もろう

# (時間)計算量

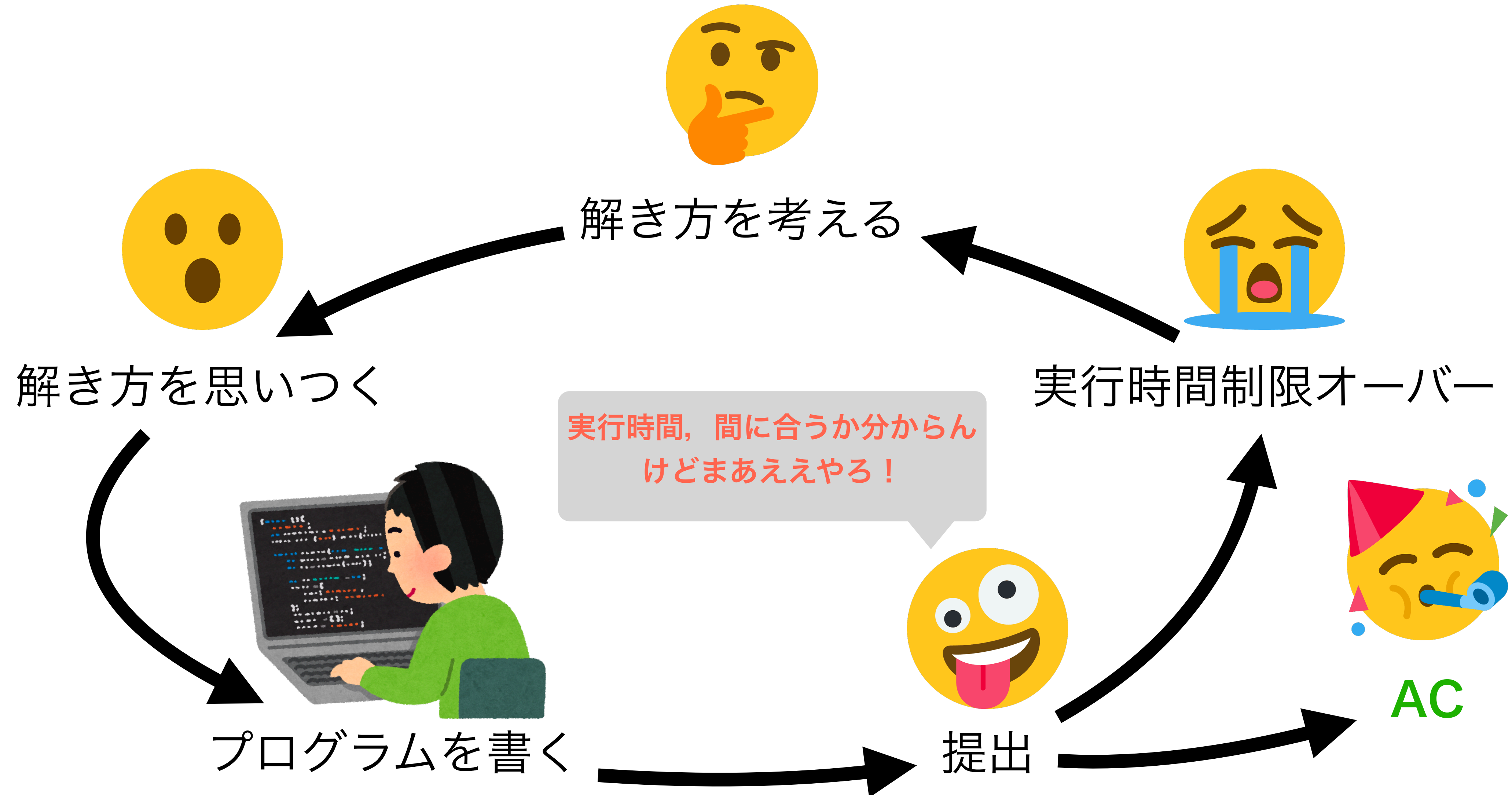
- JOI, AtCoderの問題には, 実行時間制限が定められている

時間制限 : 2 sec / メモリ制限 : 1024 MB

→ 正しくかつ高速に答えを求めるプログラムを書く必要がある

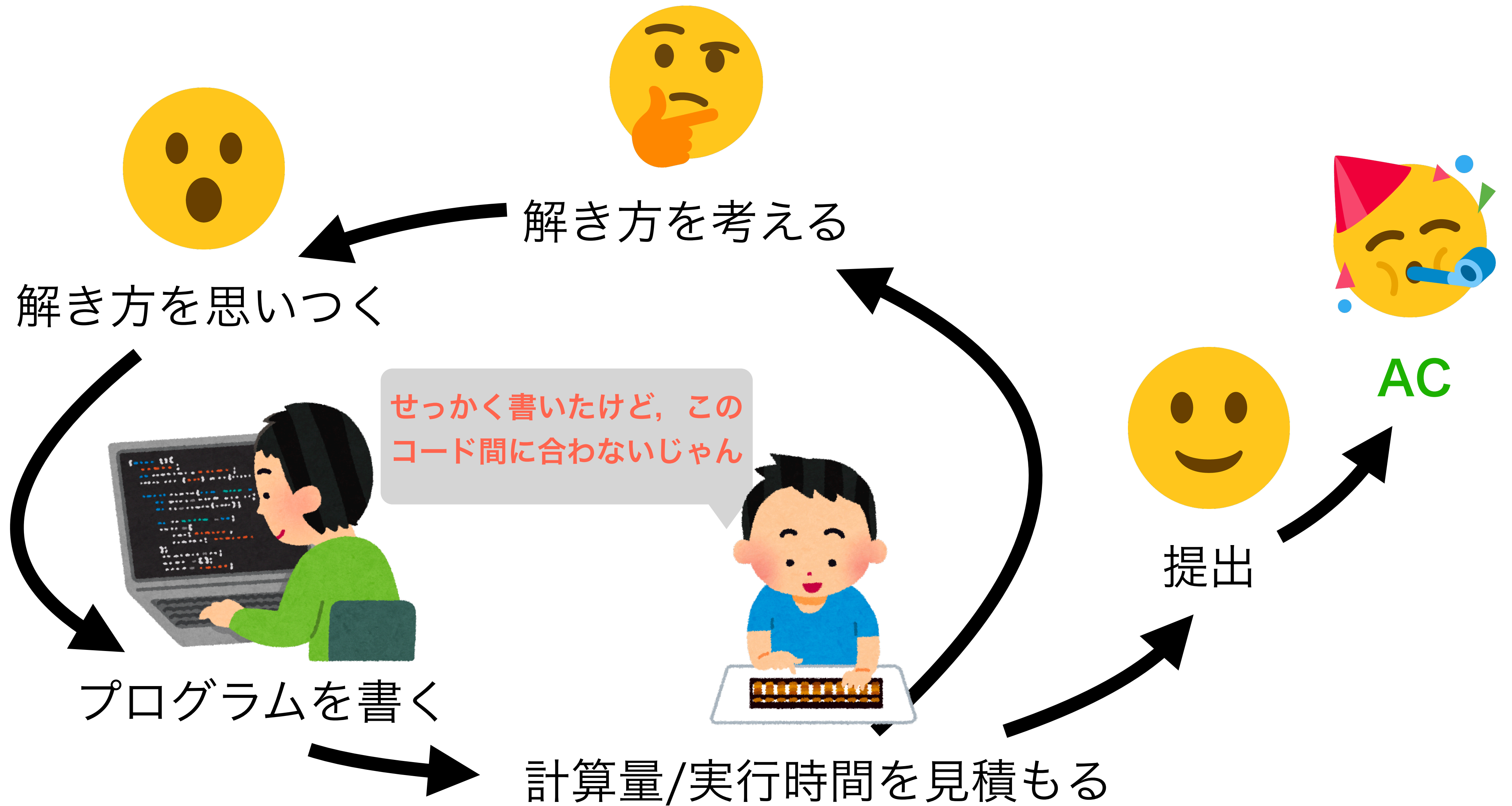
- 計算量を求めることで, プログラムの実行時間を見積もることができる

# 良くない取り組み方



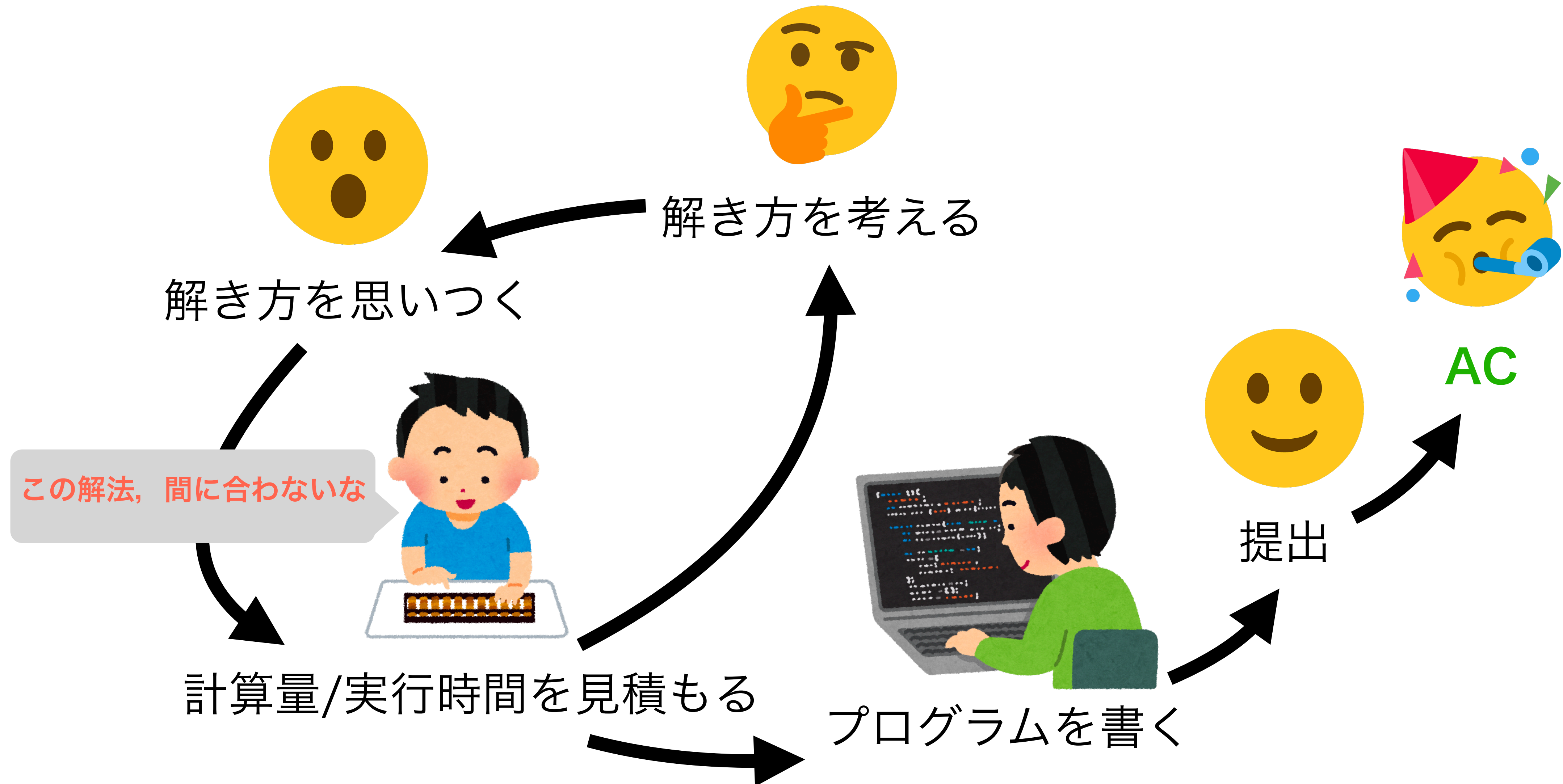


# 良い取り組み方





# とても良い取り組み方



# 1からNまでの総和を求めてみよう

## 問題文

正整数 $N$ が与えられます。1から $N$ までの総和( $1, 2, 3, \dots, N$ を全て足した値)を求めてください。

## 入力

入力は、以下の形式で与えられる。

$N$

## 出力

1から $N$ までの総和を出力せよ。

実行時間制限：500 ms

### 制約

- ・ 入力は整数
- ・  $2 \leq N \leq 4 \times 10^9$

Tips:コードテストを利用して実行時間を計測できる

終了コード	0
実行時間	1210 ms
メモリ	444 KB

← 500msを超えてる！

# 実行時間制限を超えるプログラムの例

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    unsigned long long n;
    cin >> n;

    unsigned long long ans = 0;
    for (int i = 1; i < n + 1; i++) {
        ans += i;
    }

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

標準入力

40000000000

← 制約を満たす最も大きい値

※512 KiB まで

※AtCoderの入力データでは通常、末尾にも改行が入ります

実行

終了コード

0

実行時間

1203 ms

← 500msを超えてる！

メモリ

448 KB

標準出力

80000000002000000000

← 出力は正しい

# 1からNまでの総和

$$S = 1 + 2 + 3 + \dots + (N-2) + (N-1) + N$$

$$S = N + (N-1) + (N-2) + \dots + 3 + 2 + 1$$

$$2S = (N+1) + (N+1) + (N+1) + \dots + (N+1) + (N+1) + (N+1)$$

$$2S = N \times (N+1)$$

$$S = \frac{N \times (N+1)}{2}$$

# 正解の一例

$$\sum_{i=1}^{i=N} i = \frac{N \times (N + 1)}{2}$$

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    unsigned long long n;
    cin >> n;

    unsigned long long ans = n * (n + 1) / 2;

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

標準入力

4000000000

← 制約を満たす最も大きい値

※ 512 KiB まで

※ AtCoderの入力データでは通常、末尾にも改行が入ります

実行

終了コード

0

実行時間

8 ms

← 500msより遥かに小さい！

メモリ

400 KB

標準出力

80000000002000000000

← 出力も正しい



# 計算ステップ数を数えてみよう

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    unsigned long long n;
    cin >> n;

    unsigned long long ans = 0;
    for (int i = 1; i < n + 1; i++) {
        ans += i;
    }

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

ans += i という計算は $N$ 回行われる

→ 計算ステップ数は $N$ 回

# 計算ステップ数を数えてみよう

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    unsigned long long n;
    cin >> n;

    unsigned long long ans = n * (n + 1) / 2;

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

← 計算ステップ数は1回

# 計算量

## 計算量の求め方

1. 計算ステップ数を文字式で表し, 係数を見捨てる
2. 変数を大きくしたときに, 一番影響が大きい項を取り出す
3. 計算量は  $O(\text{一番影響が大きい項})$  と表す ([https://atcoder.jp/contests/apg4b/tasks/APG4b\\_w](https://atcoder.jp/contests/apg4b/tasks/APG4b_w))

### 例

計算ステップ数が  $2N^2 + 4N + 3$  のとき,

係数を見捨ると  $N^2 + N + 1$ ,

よって, 計算量は  $O(N^2)$



# 計算量と実行時間

実行時間制限が 1~2 sec の場合の目安

$O(10^6)$	余裕を持って間に合う
$O(10^7)$	おそらく間に合う
$O(10^8)$	非常にシンプルな処理なら間に合う
$O(10^9)$	間に合わない

(プログラミングコンテストチャレンジブックより)

# 振り返り

## 問題文

正整数 $N$ が与えられます。1から $N$ までの総和( $1, 2, 3, \dots, N$ を全て足した値)を求めてください。

## 入力

入力は、以下の形式で与えられる。

$N$

## 出力

1から $N$ までの総和を出力せよ。

実行時間制限：500 ms

### 制約

- ・ 入力は整数
- ・  $2 \leq N \leq 4 \times 10^9$

# 振り返り

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    unsigned long long n;
    cin >> n;

    unsigned long long ans = 0;
    for (int i = 1; i < n + 1; i++) {
        ans += i;
    }

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

$O(N) = O(4 \times 10^9)$  ← 間に合わない！

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    unsigned long long n;
    cin >> n;

    unsigned long long ans = n * (n + 1) / 2;

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

$O(1)$

間に合う！

# 演習 1

コードの計算量を求めて、実行時間を見積もろう

# 演習 1-1

実行時間制限：2 sec  
制約

$$\cdot 0 \leq N \leq 10^5$$

- ・ 計算量は？
- ・ 実行時間は間に合う？

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N;
    cin >> N;

    int ret = 0;
    for (int i = 0; i < 3*N; i++) {
        ret += i;
    }
    cout << ret << endl;
}
```

# 演習 1-2

実行時間制限：2 sec  
制約

$$\cdot 0 \leq N \leq 10^5$$

- 計算量は？
- 実行時間は間に合う？

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N;
    cin >> N;

    int ret = 0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            ret += i + j;
        }
    }
    cout << ret << endl;
}
```

# 演習 1-3

実行時間制限：2 sec  
制約

- $0 \leq N \leq 10^5$
- $0 \leq M \leq 10^2$

- 計算量は？
- 実行時間は間に合う？

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N, M;
    cin >> N >> M;

    int ret = 0;
    for (int i = 0; i < N; i++) {
        ret += i;
    }
    for (int i = 0; i < M; i++) {
        ret += i;
    }
    cout << ret << endl;
}
```



# 演習 1-4

実行時間制限：2 sec  
制約

- $0 \leq N \leq 10^5$
- $0 \leq M \leq 10^2$

- 計算量は？
- 実行時間は間に合う？

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N, M;
    cin >> N >> M;

    int ret = 0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            ret += i + j;
        }
    }
    cout << ret << endl;
}
```



# 演習 1-5

実行時間制限：2 sec  
制約

$$\cdot 0 \leq N \leq 10^5$$

- ・ 計算量は？
- ・ 実行時間は間に合う？

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N;
    cin >> N;

    int ret = 0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < 5; j++) {
            ret += i + j;
        }
    }
    cout << ret << endl;
}
```

# 演習 1-6

実行時間制限：2 sec  
制約

- $0 \leq N \leq 10^5$
- $0 \leq M \leq 10^2$

- 計算量は？
- 実行時間は間に合う？

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N, M;
    cin >> N >> M;

    int ret = 0;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            ret += i + j;
        }
    }
    cout << ret << endl;
}
```

# 演習 1-7 (難)

実行時間制限：2 sec  
制約

$$\cdot 0 \leq N \leq 10^5$$

- ・ 計算量は？
- ・ 実行時間は間に合う？

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N;
    cin >> N;

    int ret = 0;
    while (N > 0) {
        ret++;
        N /= 2;
    }
    cout << ret << endl;
}
```

# 演習 1-8 (難)

配列のサイズを $N$ とする

関数名	計算量
sort	
find	
erase	
insert	

# 休憩！（15分）

計算量には慣れましたか？