

計算量を学ぼう！



# 計算量を学ぼう！

ぱうえる (けんた)

2022/12/20 @nu\_zero\_one

# 速いコードが書きたい！

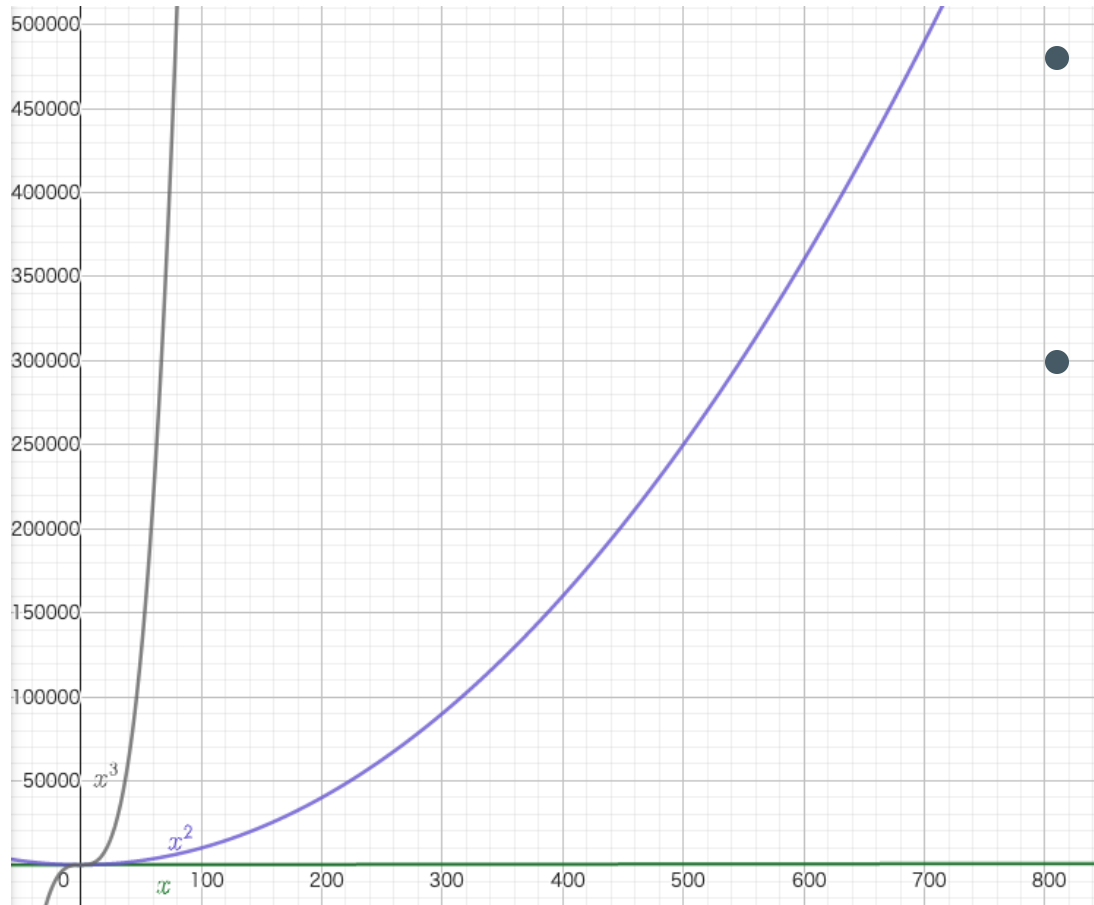
でも速いコードってどうやって評価する??

- 「1,000,000 個のデータに対して 5 秒で終了しました！」
  - データの個数が変わったらどうなる??
  - そもそもPythonで実行するかC言語で実行するかでも変わりそう

---

「データの大きさ」や「実行する環境」に依存しない評価方法が必要  
→ 計算量の出番

# オーダー記法 (1/2)



- $n, n^2, n^3$  では  $n$  が大きくなったとき  
値が大きく変化する
- 定数倍を考えないで、 $n$  の項だけに  
注目すればいいのでは??

→  $O$  (ランダウの記号)を用いる

# オーダー記法 (2/2)

- 計算量は基本的にオーダー記法で書く
  1. 一番大きい項のみ残して表記する
$$c < \log n < n^c < c^n < n! \quad (c \text{ は定数})$$
  2. 定数倍は無視する

オーダー記法の例)

$$5n^3 + 4n^2 + 100n \longrightarrow O(n^3)$$

$$2^n + n^{100} + 10^9 n \longrightarrow O(2^n)$$

# コードの計算量の調べ方

- $n$  回のループをする  $\rightarrow O(n)$
- $n$  回のループの中で  $n$  回のループをする（二重ループ）  
 $\rightarrow O(n^2)$
- bit全探索（ $n$  個の要素についてある/ないの 2 通りを考える）  
 $\rightarrow O(2^n)$
- $n$  個の順列を全て調べる  $\rightarrow O(n!)$

# ここまでの復習

このコードの計算量は？？

```
## 1~n までの数の和を求める  
n = int(input())  
  
ans = 0  
for i in range(1, n+1):  
    ans += i  
  
print(ans)
```

# ここまでの復習（答え）

このコードの計算量は？？

```
## 1~n までの数の和を求める
n = int(input())

ans = 0
for i in range(1, n+1):
    ans += i

print(ans)
```

→  $O(n)$  ( $n$  までのループを1回している)

# 計算量の使い方

- 一般的なコンピュータが1秒間に計算できる回数は約  $10^8$  回
- 競プロの実行時間制限は大体 1～3 秒
- 各計算量ごとの、制限時間に間に合う  $N$

$$O(N) \quad : N \leq 10^7$$

$$O(N \log N) \quad : N \leq 10^6$$

$$O(N^2) \quad : N \leq 10^4$$

$$O(N^3) \quad : N \leq 300$$

↓  $n$  の大きさと実際の値は次ページの表のようになります



$\log n$	$n$	$n \log n$	$n^2$	$n^3$	$2^n$	$n!$
2	<b>5</b>	12	25	130	30	120
3	<b>10</b>	33	100	1000	1024	3628800
4	<b>15</b>	59	225	3375	32768	—
4	<b>20</b>	86	400	8000	1048576	—
5	<b>25</b>	116	625	15625	—	—
5	<b>30</b>	147	900	27000	—	—
7	<b>100</b>	664	10000	1000000	—	—
8	<b>300</b>	2468	90000	27000000	—	—
10	<b>1000</b>	9966	1000000	—	—	—
13	<b>10000</b>	132877	100000000	—	—	—
16	<b>100000</b>	1660964	—	—	—	—
20	<b>1000000</b>	19931568	—	—	—	—

参考：<https://qiita.com/drken/items/872ebc3a2b5caaa4a0d0#1-3-計算量の使い方>

# 計算量を落とすテクニック

今回は代表的なものを2つ紹介します。

- **式変形**

比較的単純だけど、強力な手法

- **累積和**

数列の区間の和を高速に求めるアルゴリズム

# 式変形

先ほどの  $1 \sim n$  までの数の和を求めるプログラムを高速化してみよう

```
## 1~n までの数の和を求める
n = int(input())

ans = 0
for i in range(1, n+1):
    ans += i

print(ans)
```

# 式変形

このコードは、 $1 \sim n$  の和を求めるために  $O(n)$  の計算をしています  
( $n = 100,000,000$  で2.6秒くらい必要) →間に合わない！

```
In [6]: %%timeit
...: # 1~n までの数の和を求める
...: n = 100_000_000
...:
...: ans = 0
...: for i in range(1, n+1):
...:     ans += i
...:
```

```
2.61 s ± 3.82 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

# 式変形

等差数列の和の公式を使えば...

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

```
## 1~n までの数の和を求める  
n = int(input())  
ans = n * (n + 1) // 2  
  
print(ans)
```

# 式変形

```
In [9]: %timeit
...: # 1~n までの数の和を求める
...: n = 100_000_000
...: ans = n * (n + 1) // 2
...:
...:
53.7 ns ± 3.87 ns per loop (mean ± std. dev. of 7 runs, 10,000,000 loops each)
```

なんと、53.7ナノ秒で終了！！

- 約**5億倍**の高速化（ちょっと極端な例ではあるけど）
- もちろん余裕で間に合う

# 演習問題（式変形）

- Q4. 積の総和（1）（アルゴ式）

<https://algo-method.com/tasks/1019>

- ARC107 A - Simple Math（AtCoder）

[https://atcoder.jp/contests/arc107/tasks/arc107\\_a](https://atcoder.jp/contests/arc107/tasks/arc107_a)

# 累積和

あるたい焼き屋さんでは 毎日、売れた個数を記録しています。  
営業開始から 7 日目までの売り上げは以下の通りでした。

1日目	2日目	3日目	4日目	5日目	6日目	7日目
20	50	30	10	30	0	40

2 日目から 5 日目までの売り上げの合計はいくらでしょうか？

$$\rightarrow 50 + 30 + 10 + 30 = 120 \text{ (個)}$$



# 累積和

---

あるたい焼き屋さんでは、 $N$  日間毎日売り上げを記録しています。  
営業開始から  $i$  日目の売り上げは  $A_i$  円でした。  
このとき、以下の  $Q$  個のクエリ（質問）に答えて下さい。  
 $a$  日目から  $b$  日目までの売り上げの合計はいくらでしょうか？

---

- $1 \leq a \leq b \leq N \leq 10^5$
- $0 \leq A_i \leq 10^9$
- $1 \leq Q \leq 10^5$

# 累積和

各項を毎回足していくと、毎回のクエリで  $A_a + A_{a+1} + \dots + A_b$  という足し算をすることになる。→最大で  $O(N)$  回  
よって、 $Q$  個のクエリを処理すると、計算量は  $O(NQ)$  ！！

→間に合わない！

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$
20	50	30	10	30	0	40

# 累積和

そこで、 $S_k = \sum_{i=0}^k A_i$  を満たす  $S_i$  を考える。（累積和）  
このとき、 $S_b - S_{a-1}$  が求める区間の和になる。

証明)

$$\begin{array}{rcl} S_b & = & A_1 + A_2 + \cdots + A_{a-1} + A_a + \cdots + A_b \\ -) S_{a-1} & = & A_1 + A_2 + \cdots + A_{a-1} \\ \hline S_b - S_{a-1} & = & A_a + \cdots + A_b \end{array}$$

# 累積和

つまり??

$$50 + 30 + 10 + 30 = 140 - 20 = 120$$

$i$	0	1	2	3	4	5	6	7
$A_i$	—	20	50	30	10	30	0	40
$S_i$	0	20	70	100	110	140	140	180

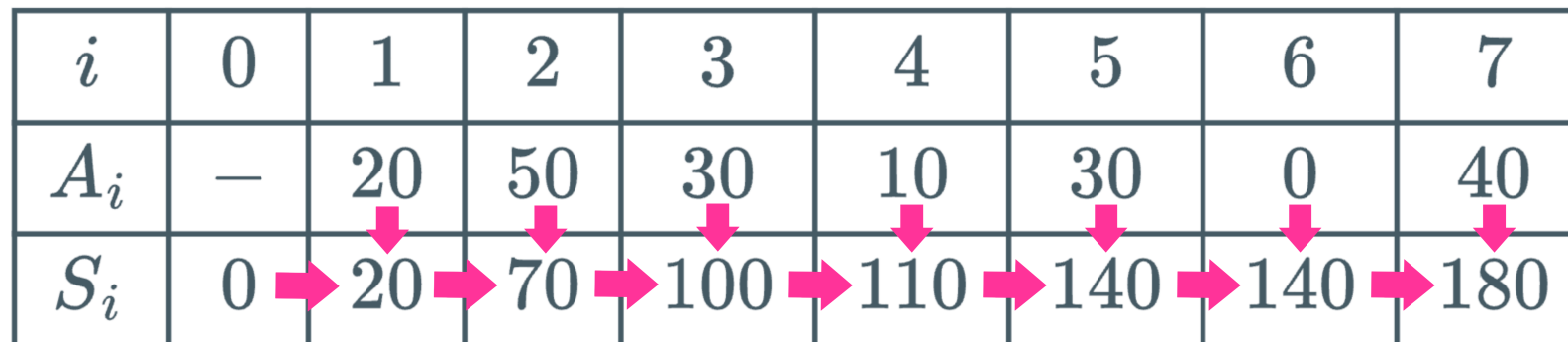
# 累積和

## 累積和配列の計算方法

$$S_0 = 0$$

$$S_i = A_i + S_{i-1} \quad (1 \leq i \leq N)$$

$i$	0	1	2	3	4	5	6	7
$A_i$	—	20	50	30	10	30	0	40
$S_i$	0	20	70	100	110	140	140	180



# 累積和

累積和の計算量は？

- $S$  を求めるのに  $O(N)$  :  $S_0$  から  $S_N$  まで  $N$  回計算する
- クエリの計算に  $O(1)$   
 $A_a$  から  $A_b$  までの和は  $S_b - S_{a-1}$  という引き算1回で求まる
- このクエリを  $Q$  回繰り返す

→  $O(N + Q)$

よって  $N = 10^5, Q = 10^5$  程度なら、余裕で間に合います。

# 演習問題（累積和）

- Q3. 駅と駅の距離（アルゴ式）

<https://algo-method.com/tasks/1027>

- ABC037 C - 総和（AtCoder）

[https://atcoder.jp/contests/abc037/tasks/abc037\\_c](https://atcoder.jp/contests/abc037/tasks/abc037_c)

# 参考

- 計算量オーダーの求め方を総整理！ ～どこからlogが出て来るか～  
<https://qiita.com/drken/items/872ebc3a2b5caaa4a0d0>
- 累積和を何も考えずに書けるようにする！  
<https://qiita.com/drken/items/56a6b68edef8fc605821>