

アルゴリズムを学ぶ

作成者

Author: のりふみ

ふらっと考える人

URL: <http://omohikane.com>

Twitter: @norio_yht

突然ですが問題です



難しく考えすぎずに答えてみてください。

月まで折り紙

地球上にいる人間の中から、見つけるとしたらこの紙(1mm)を何回折り曲げると月まで到達するでしょうか？

*月までの距離は**38万km**とする。

*紙は何回でも折り曲げることができることとする。

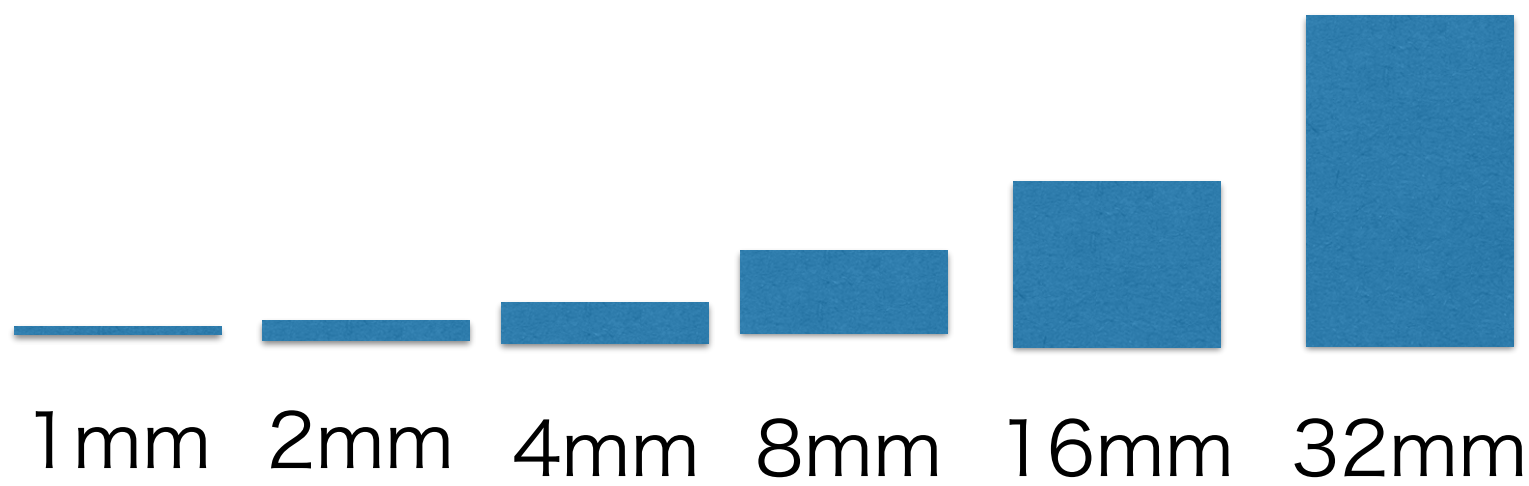


パッと答えてくださいね。

答え

39回

本当に39回でいけるのかな??



もう約1/6の回数折り曲げちゃった…

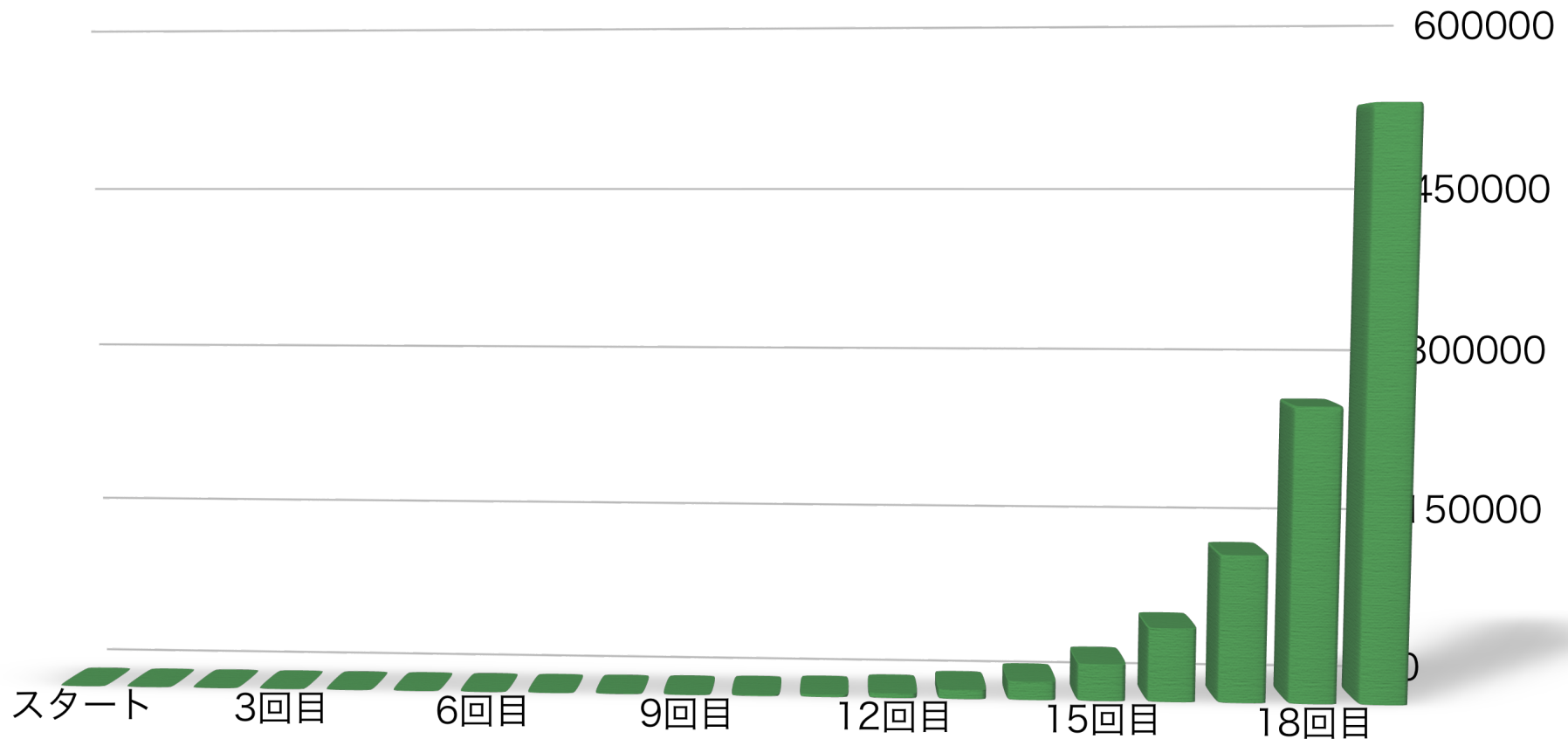
いや..これはいけちゃうぞ!!



一応並べてみると

1,2,4,8,16,32,64,128,256,512,1024,2048,4096,
8192,16384,32768,65536,131072,262144,524
288,1048576,2097152,4194304,8388608,167
77216,33554432,67108864,134217728,26843
5456,536870912,1073741824,2147483648,42
94967296,8589934592,17179869184,343597
38368,68719476736,137438953472,2748779
06944,**549755813888**

回数が少ないうちは大したことはないけど
回数が多くなると**急激**に増加する



第二問



難しく考えすぎずに答えてみてください。

犯人は誰だ

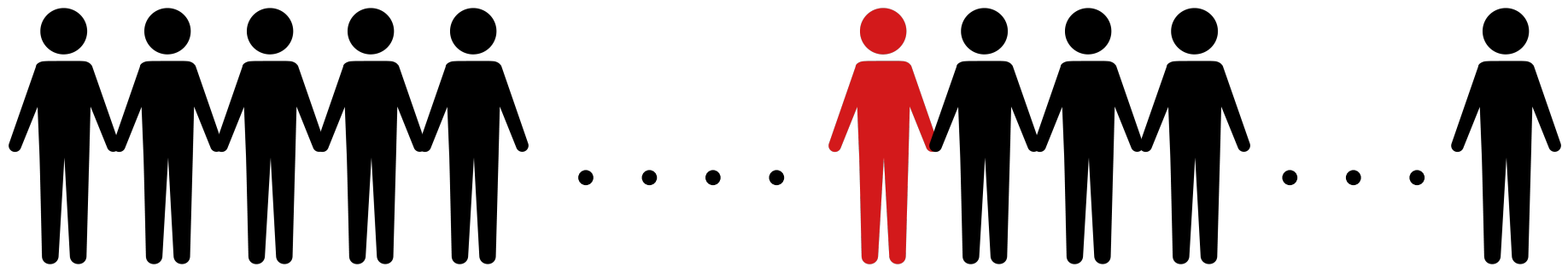
世界中の人を一行に並べ、あなたの個人情報盗んだ一人を見つけてください。犯人を見つけるためには何人に聞けばいいだろうか。

*一人には下記の三通りのどれかを答える。

自分が犯人、自分より右にいる、自分より左にいる

*世界人口は70億人いるとする

どこかにいる



7,000,000,000人

答え

最大で16回

解説の前に

多くの場合調べ方には2通りある

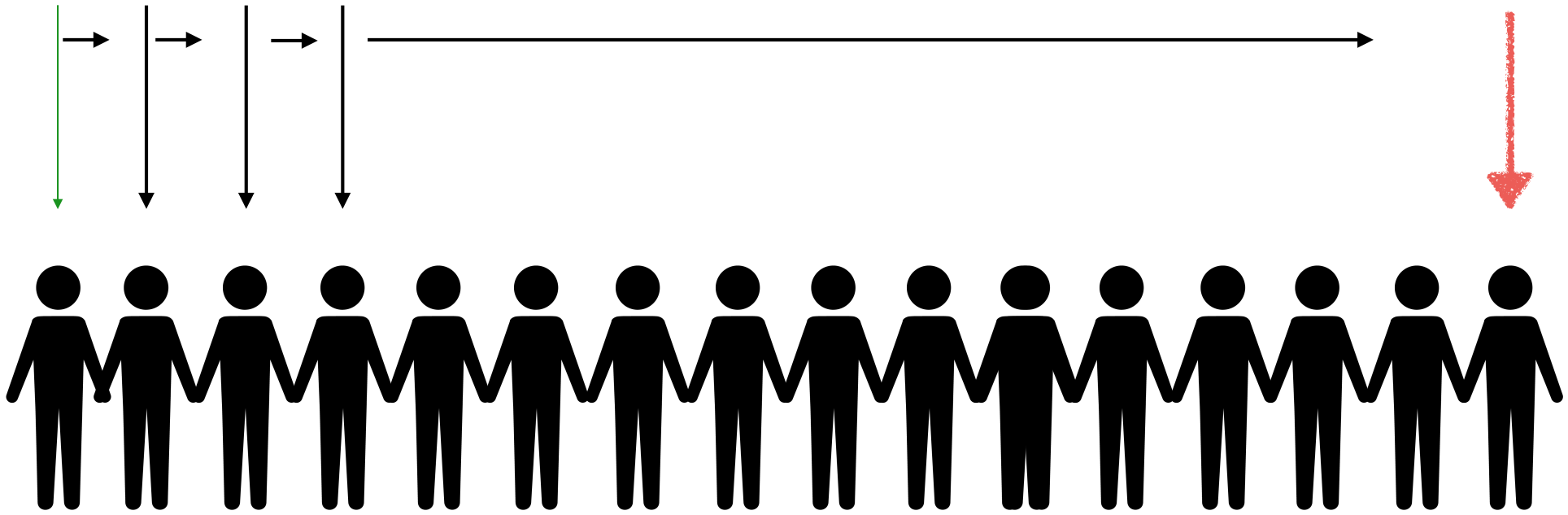
1. 「端から一人ずつ聞いていく」
2. 「真ん中の人から半分ずつ絞っていく」

1. 「端から一人ずつ聞いていく」

平均35億回で犯人を見つけられる

運が良ければ一発

運が悪ければ70億回



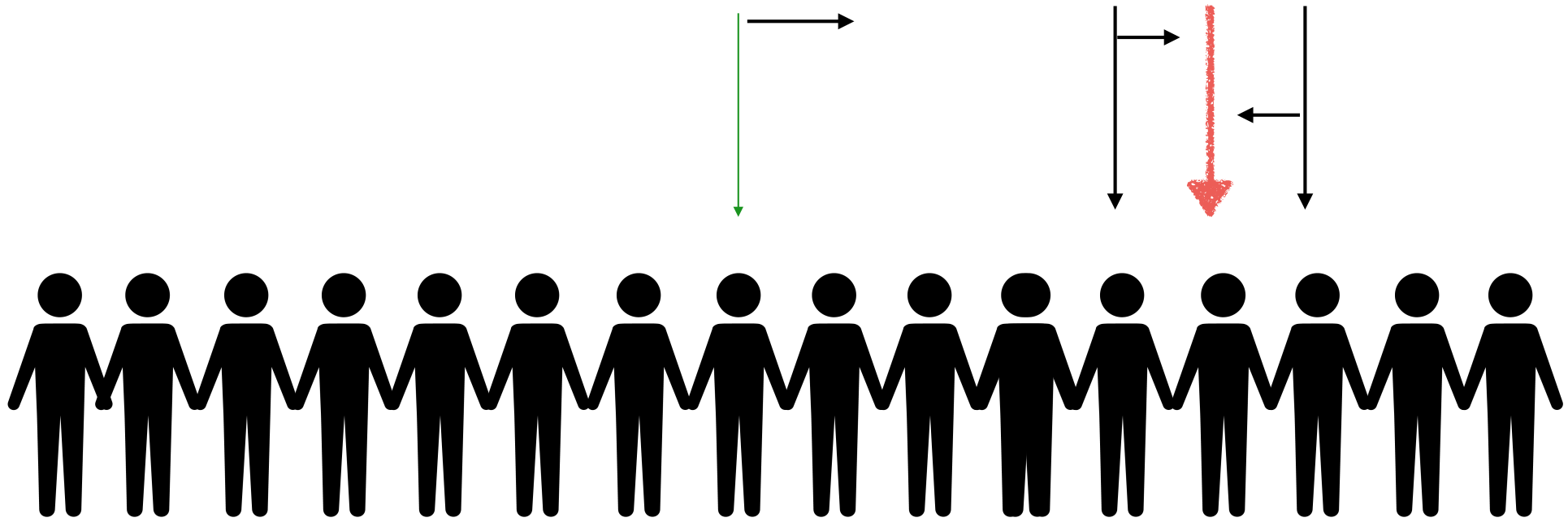
7,000,000,000人

2. 「真ん中の人から半分ずつ絞っていく」

最大で32回で犯人を見つけられる

運が良ければ一発

運が悪ければ32回



7,000,000,000人

2.の方法を使えば

どんなにかかっても32回ですむ！！

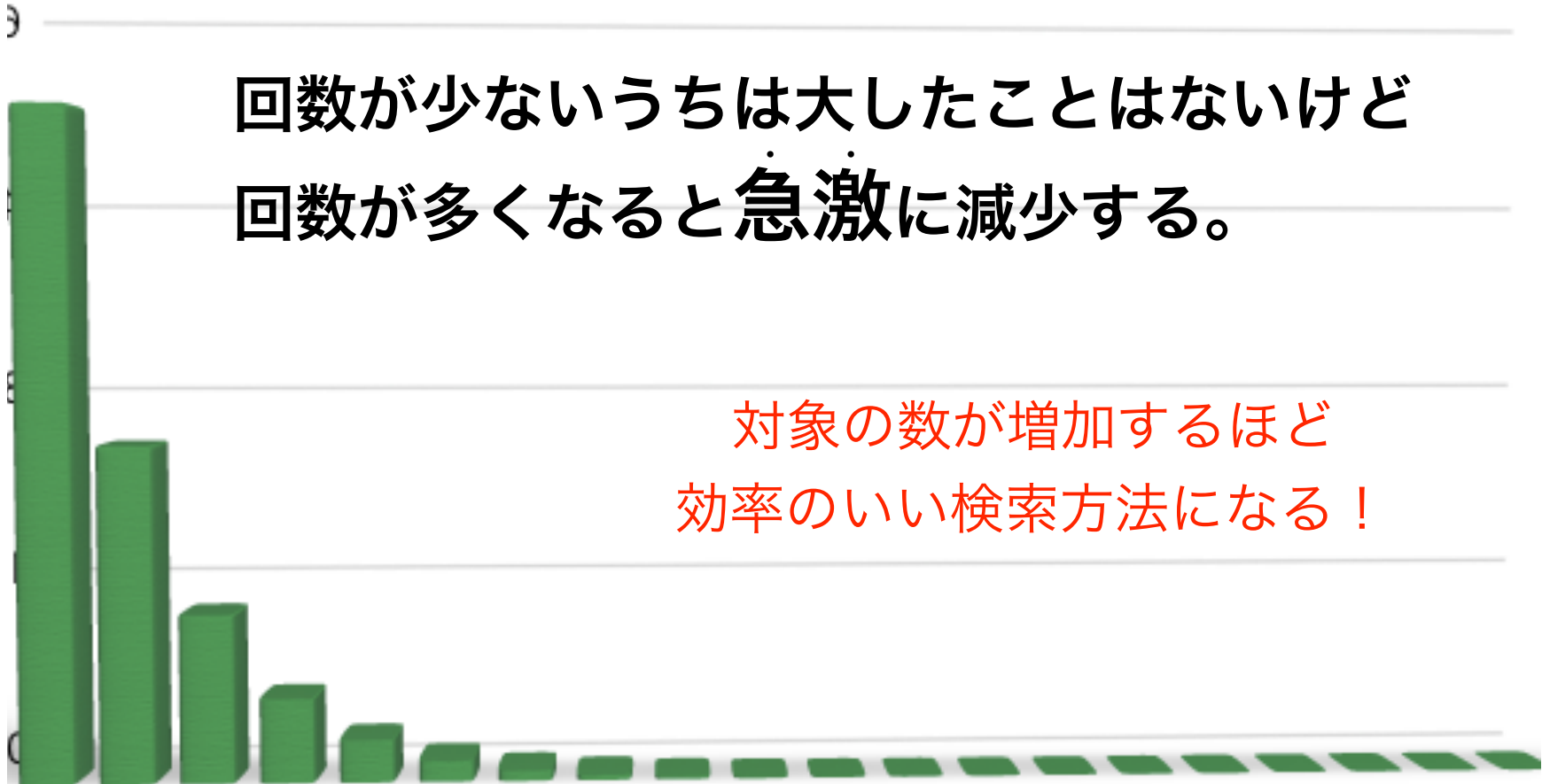


1つめの
35億回と比べたら…

2.の方法を使えば

回数が少ないうちは大したことはないけど
回数が多くなると**急激**に減少する。

対象の数が増加するほど
効率のいい検索方法になる！



2つの探し方

それぞれに
呼び方があるよ



1. 「端から一人ずつ聞いていく」

→ **Brute-force search(力任せ検索)**

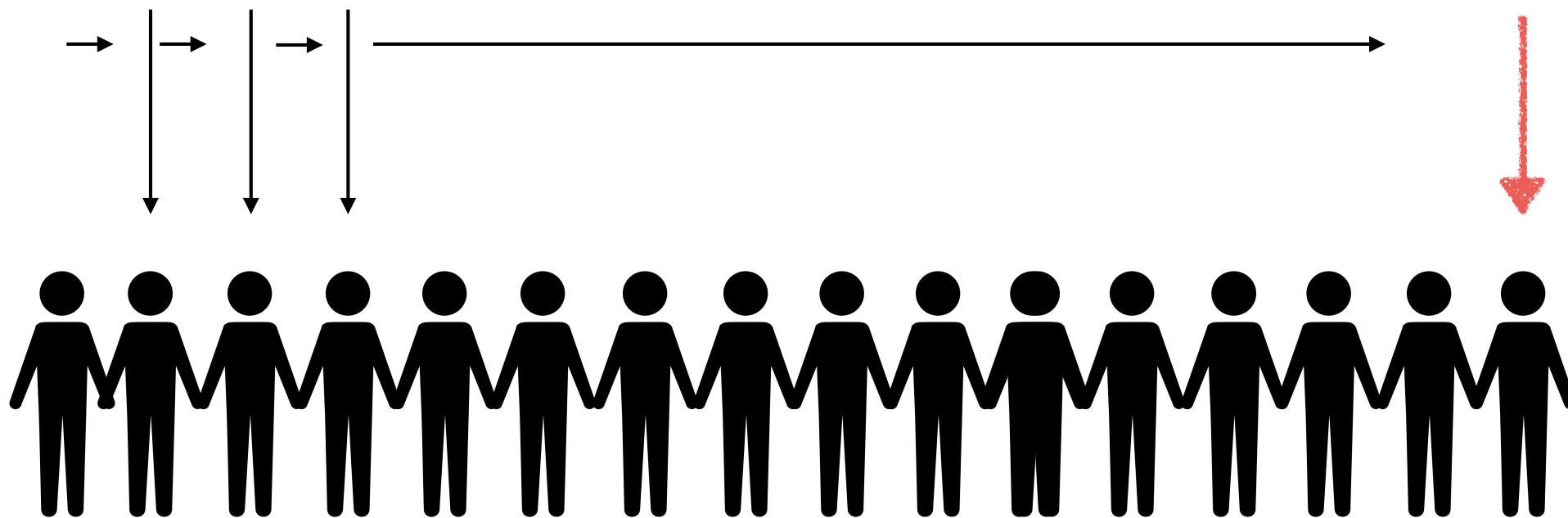
2. 「真ん中の人から半分ずつ絞っていく」

→ **Binary-search(二分法)**

Brute-force search(力任せ検索)

運が良ければ一発

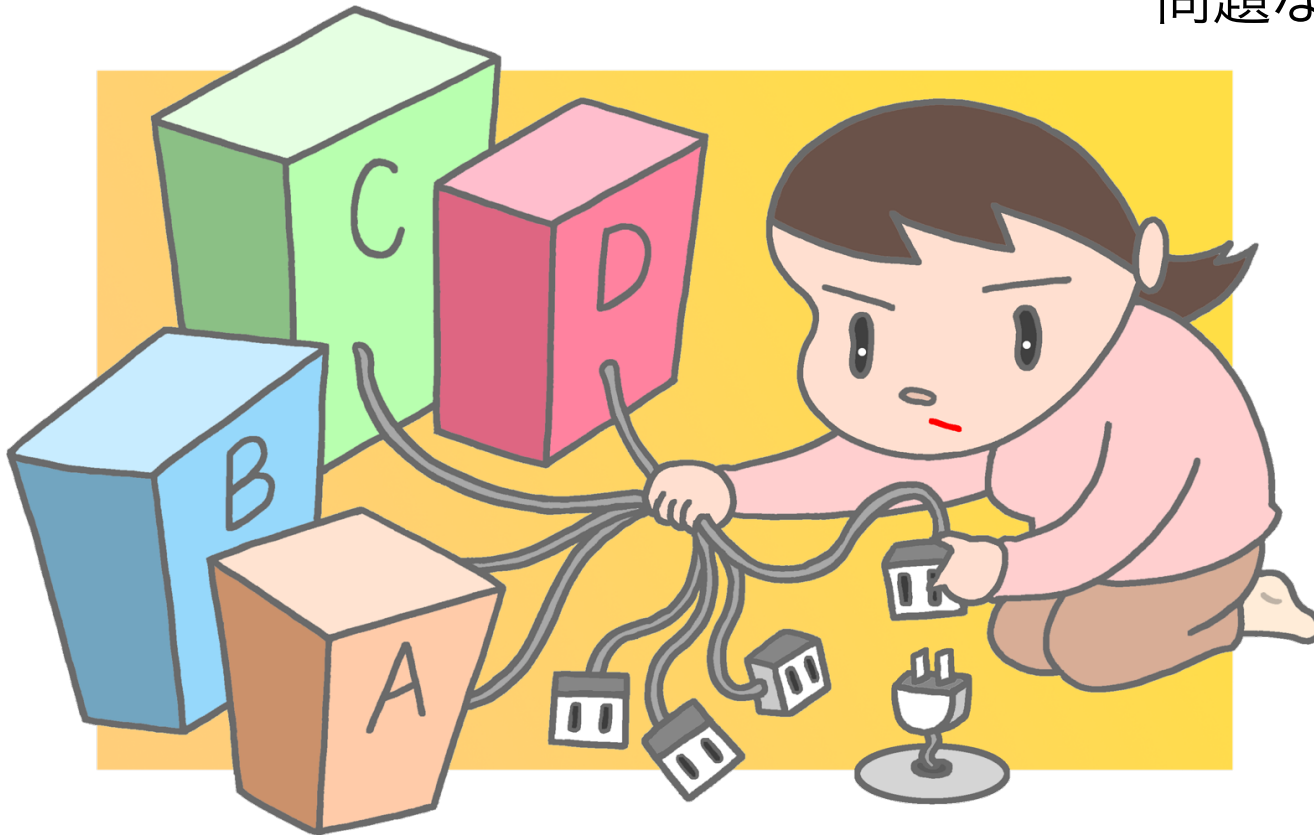
運が悪ければ70億回



7,000,000,000人

Brute-force search(力任せ検索)

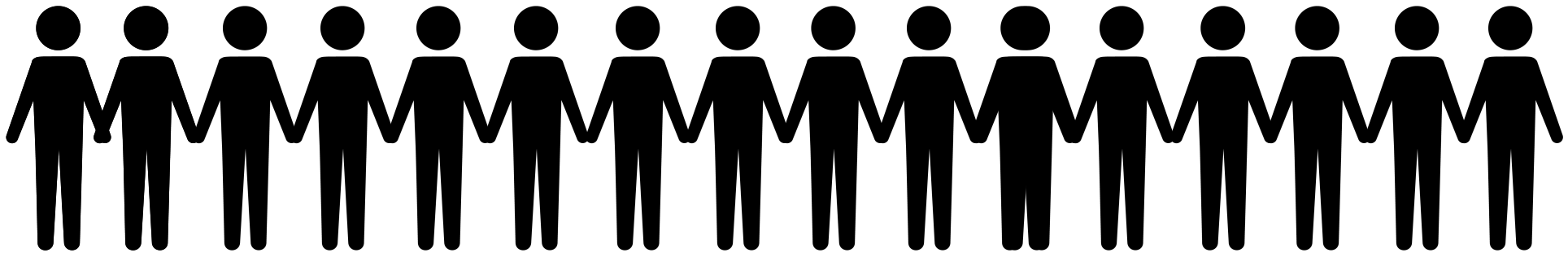
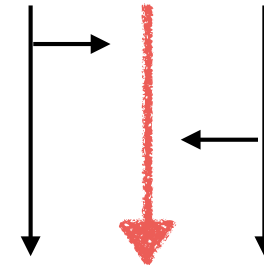
どれから調べ始めても
問題ないね



Binary-search(二分法)

運が良ければ一発

運が悪ければ32回



7,000,000,000人

Binary-search(二分法)



辞書で単語を
調べるときに似てるね

O(オーダー)

アルゴリズムの性能(複雑さ)を表す指標

O(1)

← 一番早い(ハッシュ)

O(logN)

← Binary-search

O(N)

← Brute-force-search

O(NlogN)

O(N²)

↓ Nが大きくなるとかなり遅くなる

O(2^N)

← 最悪

よりよいアルゴリズムとは

$O(1)$

$O(\log N)$

より上のオーダーを目指すこと

$O(N)$

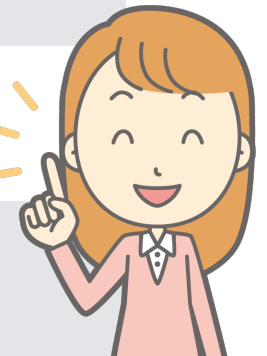
現状のオーダー

$O(N \log N)$

$O(N^2)$

そのうち詳しく説明するよ

$O(2^N)$



$O(\log N) > O(N)$ が成り立たない時があるんだ

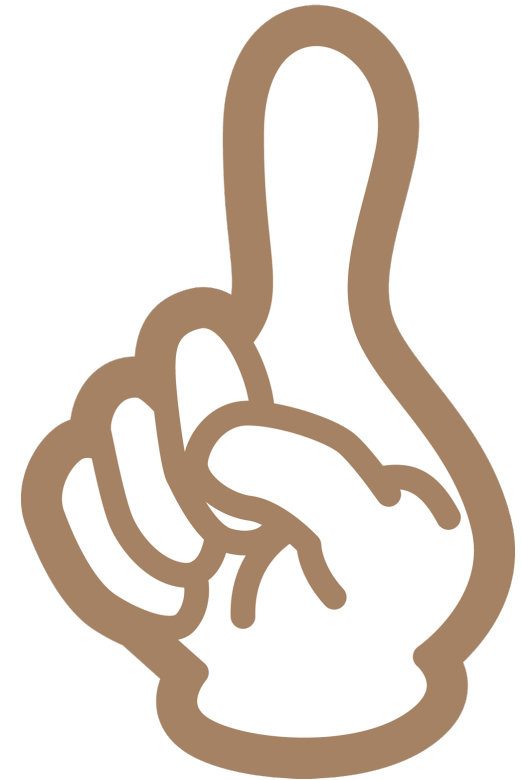
- ・ 前提条件の存在
- ・ システム要件の存在



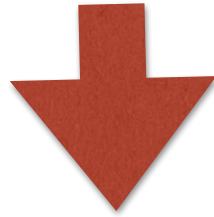
Binary-searchの前提条件

- ・ **整列している**こと

→Aの次はBでないといけない。



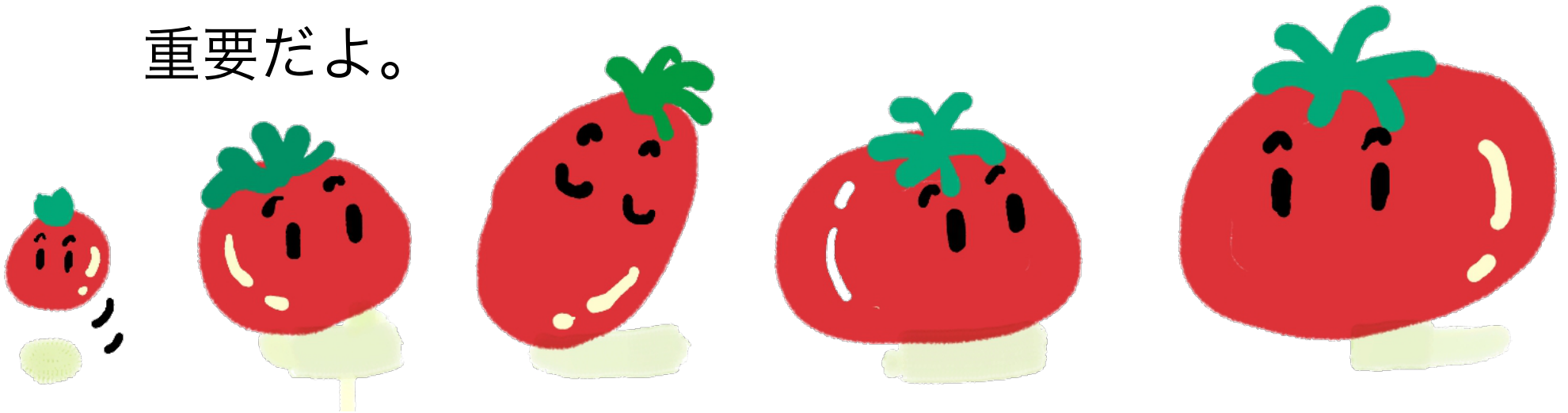
どうやって整列させるの？



ソート

これもアルゴリズム有り

重要だよ。



Brute-force-searchは？

Brute-force-searchには

→前提条件がない(ほぼ)

整列させる必要はない！！



システム要件の存在

- ・ Nの規模が極端に小さければ..

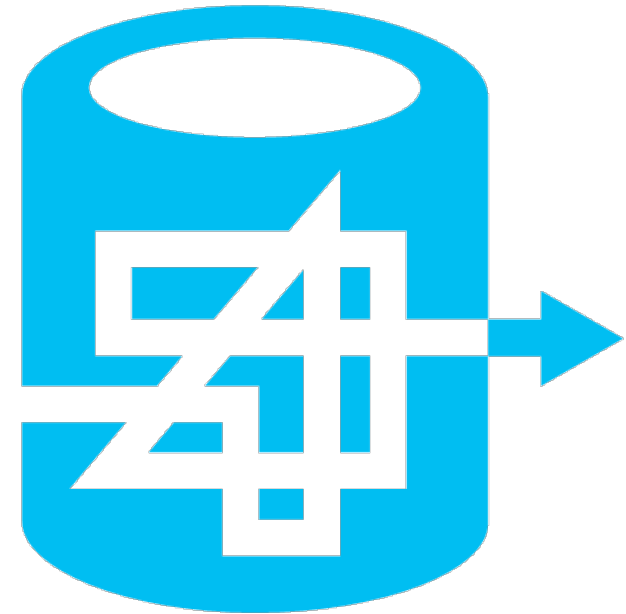
Brute-force search

でも関係ない！！



システム要件の存在

- ・ データを入力したり
- ・ データを**検索**したり
- ・ データを**編集**したり



要件によって最適なアルゴリズムが変わる

$O(\log N) > O(N)$ が成り立たない時があるんだ

今後もっと詳しく
解説するよ

- ・ 前提条件の存在
- ・ システム要件の存在



次回に向けて

- ・ 新しく知る用語ばかり
 だったけど大丈夫！

まずはC言語から
はじめよう！



作成者より

この「初心者のためのアルゴリズム入門講座」は、アルゴリズムを学びたい！という人向けに作成しております。内容に対するご指摘、ご感想はどんどん受け付けております。よりよい知識の伝播のためにもぜひご協力お願い致します。

ふらっと考える

検索

