第18章 整数論

2019/5/4 B4 hirono

JIN NAKAZAWA LABORATORY

d-hacks
JIN NAKAZAWA LAB

整数論とは？

- 整数の性質について研究する数学の分野
  - 素数とか
  - 最大公約数とか
- 情報の暗号化などの分野で大活躍！👏
- 雪江本◎

- Agenda
  - 素数判定
  - 最大公約数を求める
  - べき乗

18.1 素数判定

- **Question:** n個の整数を読み込みそれらに含まれる素数の数を出力するプログラムを作ってください

- 素数: 約数が1とその数自身だけである自然数

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	• • •			

18.1 入出力について

d-hacks
JIN NAKAZAWA LAB

入力　最初の行に n が与えられます。続く n 行に n 個の整数が与えられます。

出力　入力に含まれる素数の数を1行に出力してください。

制約　$1 \leq n \leq 10{,}000$

$2 \leq$ 与えられる整数 $\leq 10^8$

入力例

```
6
2
3
4
5
6
7
```

出力例

```
4
```

素数判定を行う素朴なアルゴリズム

- 整数xに対し2〜x-1の数で割り切れるかどうか順に調べる
  - 2で割り切れるか？
  - 3で割り切れるか？
  - …
  - n-1で割り切れるか？

全部割り切れなかったら素数！

Program 18.1: 素数判定を行う素朴なアルゴリズム

```
isPrime( x )
  if x <= 1
    return false

  for i = 2 to x-1
    if x % i == 0
      return false

  return true
```

- 全部計算しないといけない！ *計算量*$O(x)$

高速化しましょう

- 2以外の偶数は素数ではないですね→半分消える
- 　x（素数かどうか判定したい数）の半分まで調べればOK

… とやっても計算量は変わらない

- 合成数 $p \le \sqrt{x}$ を満たす素因子pを持つ という性質を利用
  - Ex. 31が素数かどうかの判定

√ 31 ≒ 5.568 → 2から6までの数で割ってみれば十分！

~~1~~	~~2~~	3	~~4~~	5	~~6~~	7	~~8~~	9	~~10~~	11	~~12~~	13	~~14~~	15	~~16~~
17	~~18~~	19	~~20~~	21	~~22~~	23	~~24~~	25	~~26~~	27	~~28~~	29	~~30~~	31	~~32~~

エラトステネスの篩（ふるい）

1. 2 以上の整数を列挙しておく
2. 最小である 2 を残して、その倍数をすべて削除
3. 残った最小の 3 を残して、その倍数をすべて削除
4. 残った最小の 5 を残して、その倍数をすべて削除
5. 以下同様に、まだ消えていない最小の数を残し、その倍数を消すことを繰り返す

1.

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60

エラトステネスの篩(ふるい)

1. 2 以上の整数を列挙しておく
2. 最小である 2 を残して、その倍数をすべて削除
3. 残った最小の 3 を残して、その倍数をすべて削除
4. 残った最小の 5 を残して、その倍数をすべて削除
5. 以下同様に、まだ消えていない最小の数を残し、その倍数を消すことを繰り返す

2.

	2	3	~~4~~	5	~~6~~	7	~~8~~	9	~~10~~	11	~~12~~	13	~~14~~	15	~~16~~	17	~~18~~	19	~~20~~
21	~~22~~	23	~~24~~	25	~~26~~	27	~~28~~	29	~~30~~	31	~~32~~	33	~~34~~	35	~~36~~	37	~~38~~	39	~~40~~
41	~~42~~	43	~~44~~	45	~~46~~	47	~~48~~	49	~~50~~	51	~~52~~	53	~~54~~	55	~~56~~	57	~~58~~	59	~~60~~

エラトステネスの篩（ふるい）

1. 2 以上の整数を列挙しておく
2. 最小である 2 を残して、その倍数をすべて削除
3. 残った最小の 3 を残して、その倍数をすべて削除
4. 残った最小の 5 を残して、その倍数をすべて削除
5. 以下同様に、まだ消えていない最小の数を残し、その倍数を消すことを繰り返す

3.

	2	3	4	5	~~6~~	7	8	~~9~~	10	11	~~12~~	13	14	~~15~~	16	17	~~18~~	19	20
~~21~~	22	23	~~24~~	25	26	~~27~~	28	29	~~30~~	31	32	~~33~~	34	35	~~36~~	37	38	~~39~~	40
41	~~42~~	43	44	~~45~~	46	47	~~48~~	49	50	~~51~~	52	53	~~54~~	55	56	~~57~~	58	59	~~60~~

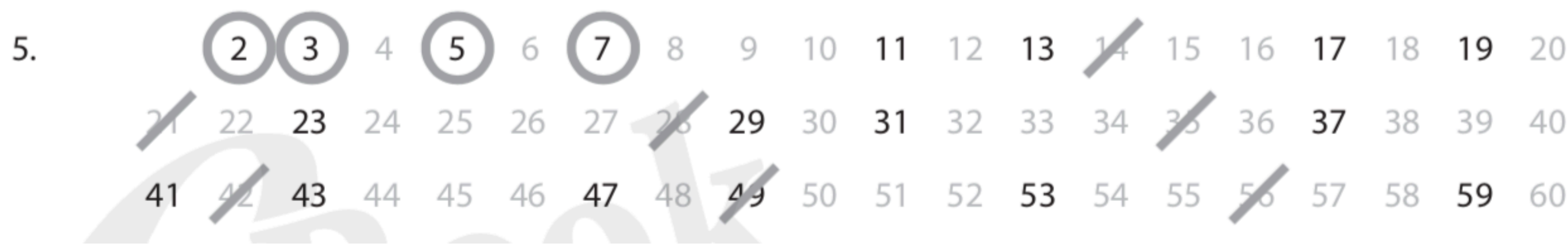
エラトステネスの篩（ふるい）

1. 2 以上の整数を列挙しておく
2. 最小である 2 を残して、その倍数をすべて削除
3. 残った最小の 3 を残して、その倍数をすべて削除
4. 残った最小の 5 を残して、その倍数をすべて削除
5. 以下同様に、まだ消えていない最小の数を残し、その倍数を消すことを繰り返す

4.		2	3	4	5	6	7	8	9	~~10~~	11	12	13	14	~~15~~	16	17	18	19	~~20~~
	21	22	23	24	~~25~~	26	27	28	29	~~30~~	31	32	33	34	~~35~~	36	37	38	39	~~40~~
	41	42	43	44	~~45~~	46	47	48	49	~~50~~	51	52	53	54	~~55~~	56	57	58	59	~~60~~

エラトステネスの篩（ふるい）

1. 2 以上の整数を列挙しておく
2. 最小である 2 を残して、その倍数をすべて削除
3. 残った最小の 3 を残して、その倍数をすべて削除
4. 残った最小の 5 を残して、その倍数をすべて削除
5. 以下同様に、まだ消えていない最小の数を残し、その倍数を消すことを繰り返す



エラトステネスの篩(ふるい)

- Program 18.3: エラトステネスの篩

```
void eratos(n)
  // 整数を列挙して素数の候補とする
  for i = 0 to n
    isprime[i] = true
  // 0 と 1 を消す
  isprime[0] = isprime[1] = false
  // i を残して i の倍数を消していく
  for i = 2 to n の平方根
    if isprime[i]
      j = i + i
      while j <= n
        isprime[j] = false
        j = j + i
```

- 調べたい整数の最大値 N に比例するメモリ領域が必要だが
- $O(N \log \log N)$ のアルゴリズム

18.2 最大公約数

- **<u>Question:</u>** 2つの自然数x, y を入力とし、それらの最大公約数を求めるプログラムを作成せよ

- 最大公約数: $x \div d$ と$y \div d$ の余りがともに0 となるd のうち最大のもの
  - Ex. 35と14の最大公約数 gcd(35, 14) = 7に！

    35の約数{1, 5, 7, 35}
    14 の約数 {1, 2, 7, 14}　公約数 { 1, 7 } の最大値

入力 x と y が1つの空白区切りで1行に与えられます。

出力 最大公約数を1行に出力してください。

制約 $1 \leq x, y \leq 10^9$

ヒント 整数x, yについて、$x \geq y$ならばxとyの最大公約数はyと$x\%y$ の最大公約数に等しい。ここで$x\%y$はx をy で割った余りである。

入力例

```
147 105
```

出力例

```
21
```

d-hacks
JIN NAKAZAWA LAB

gcdを求める素朴なアルゴリズム

- x = 35, y = 14の時 ---> 7

Program 18.5: 最大公約数を求める素朴なアルゴリズム

```
gcd(x, y)
  n = (x と y の小さい方)
  for d が n から 1 まで
    if d が x と y の約数
      return d
```

n=14

d ← 14, 13, 12, …, 1

dが35と14の約数だったらdを返す

- 最悪(素数)の場合n回の割り算を行う必要が出てくる
- 大きい数に対してはよろしくない…

ユークリッドの互除法を使い高速化

- $x \geq y$ のとき $gcd(x, y)$ と $gcd(y, x$ を y で割った余り$)$ は等しい

gcdが一緒

3355 ÷ 2379 =1 あまり 976
2379 ÷ 976 =2 あまり 427
976 ÷ 427 =2 あまり 122
427 ÷ 122 =3 あまり 61
122 ÷ 61 =2
よって最大公約数は61

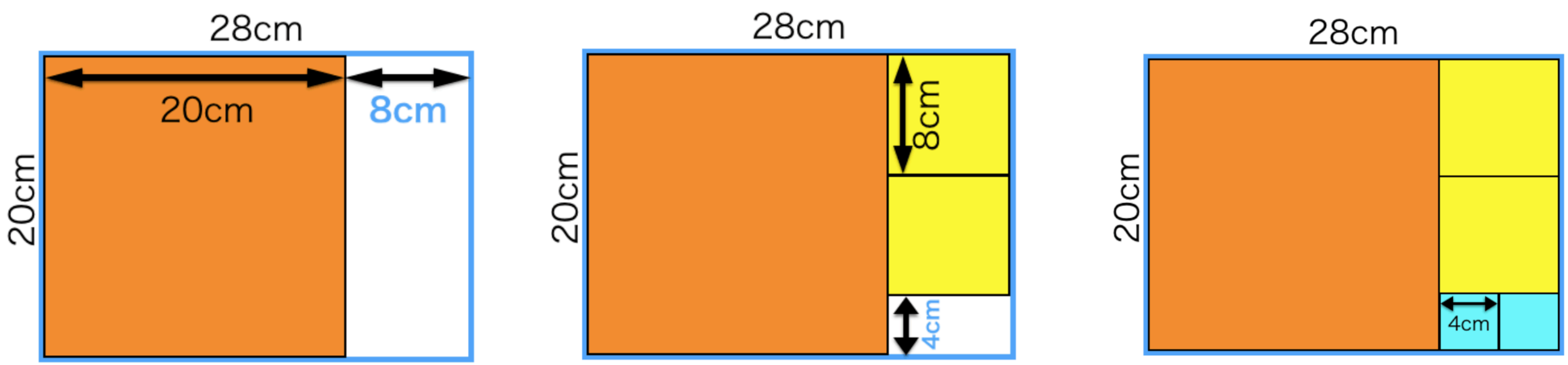
- 高校で勉強したっけ…？

補足

- gcdを求める

=x×yの長方形に敷き詰めることのできる正方形の一辺の長さdの最大値を求める！

① d = y(=20)としてみる　② d = r (=28%20=8)としてみる　③ d = r (=20%8=4) としてみる



できた！！！

[↑こちらのわかりやすい説明はこちらから↓](https://www.yukisako.xyz/entry/eucledean-algorithm)
https://www.yukisako.xyz/entry/eucledean-algorithm

ユークリッドの互除法を使い高速化

Program 18.6: ユークリッドの互除法

```
gcd(x, y)
  if x < y
    x >= y となるように x と y を交換

  while y > 0
    r = x % y       // x を y で割った余り
    x = y
    y = r

  return x
```

- 計算量について
  - $O(\log b)$

$74 = 54 \times 1 + 20 (= r_1)$

$54 = 20 \times 2 + 14 (= r_2)$

$20 = 14 \times 1 + 6 (= r_3)$

$14 = 6 \times 2 + 2 (= r_4)$

$6 = 2 \times 3 + 0 (= r_5)$

:

$b = r_1, r_2, r_3, \ldots$ の減っていき方

$a = bq + r (0 < r < b)$とすると

$r < \frac{a}{2}$ より、$r_{i+2} < \frac{r_i}{2}$:

少なくとも$2\log_2(b)$ で計算終了

18.3 べき乗

- **Question:** 2つの整数 m, n について m^n を 1,000,000,007 で割った余りを求めなさい

入力　2 つの整数 m, n が 1 つの空白区切りで 1 行に与えられます。

出力　m^n を 1,000,000,007 で割った余りを 1 行に出力してください。

制約　$1 \leq m \leq 100$

$1 \leq n \leq 10^9$

入力例

```
5 8
```

出力例

```
390625
```


- 愚直にx^nを計算すると、n-1回分の掛け算が必要 -->計算量 O(n)
- → 繰り返し自乗法を用いて高速化しよう！

繰り返し自乗法のアルゴリズム

d-hacks
JIN NAKAZAWA LAB

$$\boxed{x^n = x^{\left(2\cdot\frac{n}{2}\right)} = (x^2)^{\frac{n}{2}}}$$

これを使う！

ex. 3^{21} の計算

$$3^{21} = 3^{\left(2\cdot\frac{21}{2}\right)} = (3^2)^{\frac{21}{2}} = \underbrace{9^{10}}_{1} \underbrace{\times 3}_{2}$$

$$9^{10} = 9^{(2\cdot 5)} = (9^2)^5 = \underbrace{81^5}_{3}$$

$$81^5 = 81^{\left(2\cdot\frac{5}{2}\right)} = (81^2)^{\frac{5}{2}} = \underbrace{6561^2}_{4} \underbrace{\times 81}_{5}$$

$$6561^2 = \underbrace{6561^2 \times}_{6} 65 \ldots$$

かけ算：6回
(×)

$$\overbrace{3 \times 3 \times 3 \times \cdots \times 3 \times 3}^{n回}$$

$n-1$

かけ算：20回
(×)

- ちゃんと説明すると

$$pow(x,n) = \begin{cases} 1 & (n\text{ が0 のとき}) \\ pow(x^2, n/2) & (n\text{ が偶数のき}) \\ pow(x^2, n/2) \times x & (n\text{ が奇数のとき}) \end{cases}$$

d-hacks
JIN NAKAZAWA LAB

繰り返し自乗法のアルゴリズム

- 実装

Program 18.7: 繰り返し自乗法

```
pos(x, n)
  if n == 0
    return 1
  res = pow(x * x % M, n / 2)
  if n が奇数
    res = res * x % M
  return res
```

$$pow(x,n) = \begin{cases} 1 & (n \text{ が0 のとき}) \\ pow\,(x^2, n/2) & (n \text{ が偶数のき}) \\ pow\,(x^2, n/2) \times x & (n \text{ が奇数のとき}) \end{cases}$$

二つの整数m, nをM(=1,000,000,007)で割った余りを答える

d-hacks
JIN NAKAZAWA LAB

- 答えをM(例えば 1,000,000,007)で割った余りを求めてください」という問題では、 以下のように値を計算する

  - 足し算の場合は、加算を行うごとに％M
  - 引き算の場合は、引かれる値に M を足してから引き算を行い％M
  - 掛け算の場合は、乗算を行うごとに％M

a を M で割った余りと商をそれぞれ ar, aq

b を M で割った余りと商をそれぞれ br, bq とすると、

$$
\begin{aligned}
a \times b &= (aq \times M + ar) \times (bq \times M + br) \\
&= aq \times bq \times M^2 + ar \times bq \times M + aq \times br \times M + ar \times br \\
&= (aq \times bq \times M + ar \times bq + aq \times br) \times M + ar \times br
\end{aligned}
$$

フェルマーの小定理？

つまり

$$
\begin{aligned}
(a \times b)\%M &= ar \times br \\
&= a\%M \times b\%M
\end{aligned}
$$

18. 4 その他の問題

- Prime Factorize
  与えられた整数nを素因数分解する！
- Least Common Multiple
  与えられたn個の整数の最小公倍数を求める
- Euler’s Phi Function
  正の整数nについて1からnまでの自然数のうち
  nと互いに素なものを求める！
- Extended Euclid Algorithm
  与えられた2つの整数 a、b について
  ax + by = gcd(a, b) の解 (x, y) を求める