

ОНЛАЙН-ОБРАЗОВАНИЕ

Не забыть включить запись!



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!
Ставьте + если все хорошо

Добрый вечер!



Алгебраические алгоритмы



1. Алгоритм Евклида нахождения НОД
2. Быстрое возведение в степень
3. Решето Эратосфена
4. Быстрое вычисление чисел Фибоначчи



Алгоритм нахождения НОД (наибольшего общего делителя)

- Один из старейших алгоритмов, используемых до сих пор
- III век до нашей эры



Алгоритм Евклида

О ТУС

Алгоритм Евклида применяется к паре положительных целых чисел и формирует новую пару, которая состоит из меньшего числа и разности между большим и меньшим числом.

Процесс повторяется, пока числа не станут равными.

Найденное число и есть наибольший общий делитель исходной пары

```
пока a != b
    если a > b
        a = a - b
    иначе
        b = b - a
```

1. Найти с калькулятором НОД(125, 15)
2. Написать код за 5 минут
3. Вычислить НОД для 1234567890 и 12

Как улучшить?

O F U S

Есть проблемы?

Есть идеи?



Вместо вычитания
использовать остаток от деления

```
пока a != 0 и b != 0
    если a > b
        a = a % b
    иначе
        b = b % a
```

Вариант с рекурсией:

```
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}
```

- Иранский математик Аль-Каши, XV век
- Как вычислить x степени n быстрее чем перемножить n раз?



Быстрое возведение в степень

O T U S

Сколько нужно выполнить операций умножения, для вычисления

$$2^8$$

$$2^{10}$$

$$2^{16}$$

$$2^{31}$$

- Для ускорения использовать уже вычисленные степени
- В частности, если $n = 2^k$, то вместо $n-1$ умножения, нужно сделать k

Вариант 1

Сохраняем все промежуточные вычисления, то что нужно, используем

Вариант 2

Заранее вычисляем что нам нужно, и по ходу используем

Пример

$n = 11$
 $11 = 8 + 2 + 1$
 $x \cdot x \Rightarrow [2], [2] \cdot [2] \Rightarrow [4], [4] \cdot [4] \Rightarrow [8], [8] \cdot [2] \Rightarrow [10], [10] \cdot x \Rightarrow [11]$

Быстрое возведение в степень

O T U S

- Можно обойтись без дополнительной памяти

```
long base = 2;
long power = 10;

long res = 1;
while (power > 1) {
    if (power % 2 == 1)
        res *= base;
    base *= base;
    power /= 2;
}
if (power > 0) res *= base;

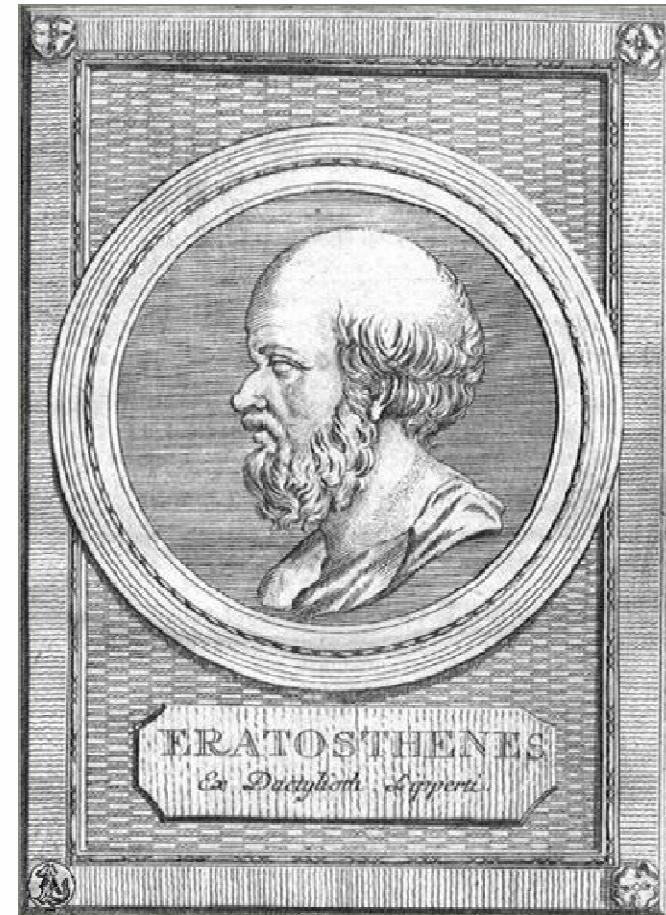
print(res);
```

Есть вопросы?

Все понятно?



Алгоритм нахождения всех простых чисел до N



Решето Эратосфена

O T U S

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Вычеркиваем каждое 2-е

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Следующее незачеркнутое число, 3, зачёркивем каждое 3-е

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Следующее незачеркнутое число, 5, зачёркиваем каждое 5-е:

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

...

- Сложность алгоритма $n \log(\log(n))$

2	3	4	5	6	7	8	9	10	Prime numbers
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	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

- Задание - написать код за 10 минут
- Создать bool prime[N], заполнить true.
- Организовать цикл j от 2 до N с шагом 1
 - Только для prime[j] == true
 - Цикл k от $j*j$ до N с шагом j
 - prime[k] = false

- Обрабатывать только нечетные числа
- Начинать просеивать с j^2
- Заканчивать когда $j^2 > N$
- Хранить по 32 значения в каждом элементе массива BitSet

- Число фибоначчи

$$F(n) = F(n-1) + F(n-2)$$

и $F(1) = F(2) = 1$.

3: $1 + 1 = 2$

4: $1 + 2 = 3$

5: $2 + 3 = 5$

6: $3 + 5 = 8$

7: $5 + 8 = 13$

8: $8 + 13 = 21$

9: $13 + 21 = 34$

10: $21 + 34 = 55$

11: $34 + 55 = 89$

...

и т. д.



рекурсия

```
функция f(n)
если n < 2
    вернуть 1
иначе
    вернуть f(n-1) + f(n-2)
```

Динамическое программирование

```
f1 = 1
f2 = 1

для i от 3 до N
    f3 = f1 + f2
    f1 = f2
    f2 = f3
```

Рекуррентная формула через золотое сечение

$$x^n = F_n = F_{n-1} + F_{n-2} = x^{n-1} + x^{n-2},$$

$$\phi = \frac{1 + \sqrt{5}}{2}$$

$$F_n = \left\lfloor \frac{\phi^n}{\sqrt{5}} + \frac{1}{2} \right\rfloor.$$

Задание - написать код за 10 минут

- золотое сечение
- динамическое программировани
- Вычислить для $N = 1000$
- Поставить ! в чат кто будет готов

Матричный алгоритм

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix}.$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}.$$

- Алгоритм Евклида НОД
- Быстрое возведение в степень
- Решето Эратосфена поиска простых чисел
- Быстрое вычисление чисел Фибоначчи



Числа Фибоначчи через перемножение матриц



**Заполните, пожалуйста,
опрос о занятии**



Спасибо
за внимание!

