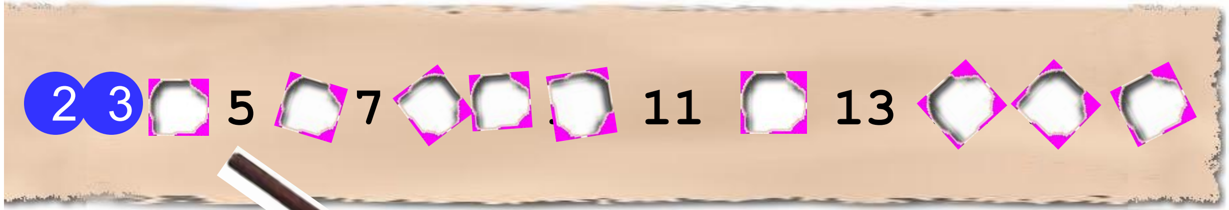
Алгоритмизация и программирование. Язык C++

§ 38. Целочисленные алгоритмы

Решето Эратосфена



Эратосфен Киренский
(Eratosthenes, Ερατοσθένης)
(ок. 275-194 до н.э.)

Алгоритм:

1) начать с **k = 2**
2) «выколоть» все числа через **k**, начиная с **k·k**
3) перейти к следующему «невыколотому» **k**
4) если **k·k <= N** , то перейти к шагу 2
5) напечатать все числа, оставшиеся «невыколотыми»

Новая версия – решето Аткина.

? Как улучшить?

\+ высокая скорость, количество операций

$$O((N \cdot \log N) \cdot \log \log N)$$

− нужно хранить в памяти все числа от **1** до **N**

Решето Эратосфена

Задача. Вывести все простые числа от 2 до N.

Объявление переменных:

```
const int N = 100;
bool A[N+1];
int i, k;
```

выделяем на 1 элемент больше, чтобы начать с **A[1]**

Сначала все невычеркнуты:

```
for ( i = 2; i <= N; i++ )
  A[i] = true;
```

Решето Эратосфена

Вычёркивание непростых:

```
k = 2;
while ( k*k <= N ) {
  if ( A[k] ) {
    i = k*k;
    while ( i <= N )
      {
      A[i] = false;
      i += k;
      }
    }
  k ++;
  }
```

Решето Эратосфена

Вывод результата:

```
for ( i=2; i<=N; i++ )
  if ( A[i] )
    cout << i << " ";
```

Наибольший общий делитель (GCD)

- gcd(a,b): наибольшее целое число делящее нацело a и b
- Используется очень часто в различных теоретических проблемах

- Несколько фактов:
  - gcd(a,b)=gcd(a,$b-a$)
  - gcd(a,0)=a
  - gcd(a,b) наименьшее положительное число в $\{ax+by \mid x,y \in Z\}$

Алгоритм Евклида

- Повторение gcd(a,b)=gcd(a,$b-a$)
- gcd(1989,867) =
  - gcd(1989−2×867,867) =
  - gcd(255,867) =
  - gcd(255,867−3×255) =
  - gcd(255,102) =
  - gcd(255−2×102,102) =
  - gcd(51,102) =
  - gcd(51,102−2×51) =
  - gcd(51,0) =51

Реализация

```
int gcd(int a, int b) {
   while(b){
      int r = a % b; a = b; b = r;
   }
   return a;
}
```

- Сложность: $O(\log(a+b))$
- Надо быть осторожным: a % b зависит от знака a
  - `5 % 3 == 2`
  - `-5 % 3 == -2`