Студент: Маргарита Чудова
Группа: 2
Дата: 27 апреля 2022 г.

---

1. Алгорим Евклида выглядит следующим образом:

```
def gcd(a,b):
    if b == 0:
        return a
    return gcd(b, a mod b)
```

Рассмотрим какие- то числа a и b. Предположим, что $a < b$ — тогда первая итерация алгоритма просто поменяет эти числа местами, выдаст (b, a). На второй итерации на выходе алгоритма мы получим $(a, b \bmod a)$, $b \bmod a < a$.
То есть начиная с этого момента мы просто работаем алгоритмом Евклида с двумя числами $\leq$: мы на каждой итерации берем остаток то от одного, то от другого числа $\leq a$, меняем их местами.

На лекции мы доказывали, что на каждом шаге алгоритма Евклида большее число уменьшается хотя бы в 2 раза. Действительно, если мы рассмотрим $\gcd(x, y)$, то у нас есть два варианта (считаем, что $x > y$): $x \geq 2y$, тогда $x\%y < y \leq x/2$ или $x < 2y$, тогда можем вычесть y только один раз и остаток равен $r = x - y < x/2$, $x\%y < x/2$.

Чисел, с которыми мы работаем, всего 2, и получается, что до конца алгоритма эти операции займут $\mathcal{O}(\log a) + 1$ времени — единица появляется, потому что когда дойдем до 0, нужна будет еще одна операция на сравнение.
Учтем еще и первую, на которой числа могут просто поменяться местами, и вспомним, что любое из чисел $a,\ b$ может быть минимальным. Тогда как раз получим $\mathcal{O}(\log(\min(a, b) + 1) + 1)$

2. (a) Нужно будет учесть тот факт, что операция $a \bmod b$ больше не выполняется за $\mathcal{O}(1)$. Она по сути своей делается через деление в столбик, что занимает $\mathcal{O}(nm)$. При этом из предыдущей задачи знаем, что асимптотика с $\mathcal{O}(1)$ будет $\mathcal{O}(\log(\min(a, b) + 1) + 1)$. Тогда асимптотика нашей задачи $\mathcal{O}(mn \log(\min(a, b) + 1) + 1) = \mathcal{O}(mn \min(m, n))$

   (b)
   - Это утверждение эквивалентно следующему

     $$\gcd(2a, 2b) = 2\gcd(a, b)$$

     Необходимость: $2a \vdots \gcd(2a, 2b), 2b \vdots \gcd(2a, 2b) \Rightarrow a \vdots \frac{1}{2}\gcd(2a, 2b),\ b \vdots \frac{1}{2}\gcd(2a, 2b)$

     Достаточность: $a \vdots \gcd(a, b), b \vdots \gcd(a, b) \Rightarrow 2a \vdots 2\gcd(a, b),\ 2b \vdots 2\gcd(a, b)$
     Итого соответствующие НОД являются делителями чисел из другой части уравнения. Заметим, что это работает не только для НОД, а для любых делителей — значит, множества общих делителей $a,\ b$ и $2a,\ 2b$ совпадают с точностью до $1/2$. Значит, и максимумы в этих множествах совпадают.
   - Это утверждение эквивалентно следующему (a нечетное)

     $$\gcd(a, 2b) = \gcd(a, b)$$

Необходимость: $a \vdots d,\ 2b \vdots d \Rightarrow a \vdots d, b \vdots \frac{d}{2}$

Достаточность: $a \vdots d,\ b \vdots d \Rightarrow a \vdots d, 2b \vdots d$
Получается, что множество общих делителей a, b и $a, 2b$ совпадает, значит и их максимумы (НОД) совпадают.

Учитывая эти факты, можно проверять числа на четность/нечетность и завести отдельную переменную k, в которую мы будем складывать все неучтенные двойки от четных чисел.

Построим алгоритм за $\mathcal{O}(max(n, m)^2)$

Заведем счетчик $k = 1$. На вход подаются числа $a,\ b$
До тех пор, пока одно из чисел не станет нулем (как только станет, берем второе за ответ):
1. Проверяем $a,\ b$ на четность.

а) Если они оба четны, то умножаем k на 2, и делим и a, и b пополам за $\mathcal{O}(max(n, m)$, возвращаемся к проверке для новых a, b.
б) Если одно четно, другое нечетно, то делим четное пополам за $\mathcal{O}(max(n, m))$, с k и нечетным ничего не делаем, возвращаемся к проверке.
в) Если оба нечетные, переходим к пункту 2.

2. Оба числа оказались нечетными, вместо взятия остатка, как в обычном алгоритме Евклида, вычтем одно из другого за $\mathcal{O}(max(n, m))$. Вычитая из нечетного нечетное, получим четное, поэтому сможем перейти к пункту 1 и уменьшить одно из чисел в два раза.

В конце домножаем ответ на k (если честно, не поняла комментарий про 2^k. Если я все правильно понимаю, так было бы, если бы встретив два четных числа, мы бы прибавляли к k единицу, а мы сразу умножаем его на 2 :))
За счет того, что на каждом шаге хотя бы одно из чисел уменьшается в 2 раза, шагов будет не больше, чем $\mathcal{O}(max(n, m))$, на каждом шаге мы делаем операций на ту же асимптотику, поэтому построили алгоритм за $\mathcal{O}(max(n, m)^2)$

3. Этот код отличается от кода, который был на лекции только тем, что второй цикл начинается с $j = i * i$, а не с $j = 2 * i$.
   Для того, чтобы доказать корректность, будем доказывать по индукции следующее утверждение: в решете Эратосфена все значения, меньшие i^2 и делящиеся на i уже были выколоты на предыдущих итерациях цикла.

   База: $i = 2, 3$ – очевидно, $i = 4 \Rightarrow i^2 = 16$; 4, 8, 12 уже выколоты двойкой.

   Переход: рассмотрим $i + 1$ итерацию.

   Пусть существует число $x = q_1^{\beta_1} \dots q_n^{\beta_n}(i + 1) < (i + 1)^2$, оно делится на $i + 1$, но не равно $i + 1$. По индукционному предположению числа, кратные чему-то меньшему $i + 1$, либо уже выколоты, либо просты и на $i+1$ делиться не могут. Тогда среди q_i должно быть хотя бы одно число, большее $i+1$, . Но тогда $x > (i+1)^2$, значит, мы пришли к противоречию, и индукционное предположение верно.