

Простые алгоритмы

Мячин Данил, ВШЭ ФКН ПМИ

Системы счисления

Давайте произведём перевод из 10-ной в 2-ную
числа 106



Давайте попробуем
запрограммировать это
всё

(кодом я поделюсь, но, пожалуйста, дома пишите
свой, чтобы оттачивать своё кунг-фу)

Наибольший общий делитель

Наибольший общий делитель

Задача: есть два числа a и b . Нужно найти такое число g , что a делится на g , b делится на g , и при этом g максимально.

Наибольший общий делитель

Задача: есть два числа a и b . Нужно найти такое число g , что a делится на g , b делится на g , и при этом g максимально.

Наивный алгоритм - перебрать все числа от 1 до $\min(a, b)$ и взять наибольшее.

Посчитаем асимптотику

Посчитаем асимптотику

$$O(\min(a,b))$$

Люди научились находить НОД (gcd) быстрее!

Да ещё и попроще

Есть ещё одна простая и клёвая идея, которую
бы я сегодня хотел рассказать

Суффиксальный массив сумм

Суффиксальный массив сумм

Звучит сложно, а на деле легко

Суффиксальный массив сумм

Звучит сложно, а на деле легко

- был вектор x , в котором n элементов

Суффиксальный массив сумм

Звучит сложно, а на деле легко

- был вектор x , в котором n элементов
- Давайте заведём вектор `suf_mas` и в i -е значение, это $x[0] + x[1] + \dots + x[i]$

Суффиксальный массив сумм

Звучит сложно, а на деле легко

- был вектор x , в котором n элементов
- Давайте заведём вектор `suf_mas` и в i -е значение, это $x[0] + x[1] + \dots + x[i]$
- Сколько займёт вычисление? Если вычислять только одно значение `suf_mas[i]` каждый раз, то каждый раз будет за $O(n)$

Суффиксальный массив сумм

Звучит сложно, а на деле легко

- был вектор x , в котором n элементов
- Давайте заведём вектор `suf_mas` и в i -е значение, это $x[0] + x[1] + \dots + x[i]$
- Сколько займёт вычисление? Если вычислять только одно значение `suf_mas[i]` каждый раз, то каждый раз будет за $O(n)$
- Но мы можем по порядку вычислять, сначала `suf_mas[0]`, потом `suf_mas[1]`, потом `suf_mas[2]`

Суффиксальный массив сумм

Звучит сложно, а на деле легко

- был вектор x , в котором n элементов
- Давайте заведём вектор `suf_mas` и в i -е значение, это $x[0] + x[1] + \dots + x[i]$
- Сколько займёт вычисление? Если вычислять только одно значение `suf_mas[i]` каждый раз, то каждый раз будет за $O(n)$
- Но мы можем по порядку вычислять, сначала `suf_mas[0]`, потом `suf_mas[1]`, потом `suf_mas[2]`
- заметим вот что (почему мы можем все значения сразу за $O(n)$ вычислить)

- $\text{suf_mas}[0] = x[0]$
- $\text{suf_mas}[1] = x[0] + x[1]$
- $\text{suf_mas}[2] = x[0] + x[1] + x[2]$
- $\text{suf_mas}[3] = x[0] + x[1] + x[2] + x[3]$
- ...
- $\text{suf_mas}[n - 1] = x[0] + x[1] + x[2] + \dots + x[n - 1]$

- $\text{suf_mas}[0] = x[0]$
- $\text{suf_mas}[1] = \text{suf_mas}[0] + x[1]$
- $\text{suf_mas}[2] = \text{suf_mas}[1] + x[2]$
- $\text{suf_mas}[3] = \text{suf_mas}[2] + x[3]$
- ...
- $\text{suf_max}[n - 1] = \text{suf_max}[n - 2] + x[n - 1]$

Что мы получили и зачем это нам?

- Один раз предобработав массив за $O(n)$ мы можем отвечать на запросы вида “Сумма элементов с 0 по k ” за $O(1)$ (просто выведем `suf_mas[k]`)
- Мы можем даже научиться отвечать на запросы вида “Сумма элементов с i по j ” за $O(1)$.
- Как? Ну давайте посмотрим.

Давайте посмотрим на i и j

- Если мы просто выведем `suf_mas[j]`, то у нас будут лишние элементы в сумме
- А какие это элементы?
- Это $x[0] + x[1] + \dots + x[i - 1]$
- Так а может их просто вычесть? Ведь эту сумму мы тоже за $O(1)$ умеем находить!
- Вуаля! Умеем решать эту задачу за $O(1)$

Если успеем, пройдемся по ошибкам во втором
дз