Рекурсивные алгоритмы

Мячин Данил, ВШЭ ФКН ПМИ

Но перед этим

Давайте подумаем, как проверять числа на простоту

Давайте подумаем, как проверять числа на простоту

- Задача – дано число n. Нужно проверить, является ли n простым.

Давайте подумаем, как проверять числа на простоту

- Задача – дано число n. Нужно проверить, является ли n простым.

Тривиальный алгоритм – перебрать все числа от 1 до n и проверить n на делимость всех этих чисел

Давайте подумаем, как проверять числа на простоту

- Задача – дано число n. Нужно проверить, является ли n простым.

Тривиальный алгоритм – перебрать все числа от 2 до n-1 и проверить n на делимость всех этих чисел

Асимптотика – O(n) (мы проверим каждое из чисел от 1 до n)

Можно сделать чуть умнее, но сначала докажем один факт

Можно сделать чуть умнее, но сначала докажем один факт

- Если число n представимо в виде произведения a * b, то если a < b,

Можно сделать чуть умнее, но сначала докажем один факт

- Если число n представимо в виде произведения a * b, то если a < b,
- a <= sqrt(n)

Можно сделать чуть умнее, но сначала докажем один факт

- Если число n представимо в виде произведения a * b, то если a < b,
- a <= sqrt(n)
- b >= sqrt(n)

Можно сделать чуть умнее, но сначала докажем один факт

- Если число n представимо в виде произведения a * b, то если a < b,
- a <= sqrt(n)
- b >= sqrt(n)

Докажем?

- Предположим, что $a < \mathrm{sqrt}(n)$
- $\mathrm{sqrt}(n) = n \;/\; \mathrm{sqrt}(n)$
- $b = n \;/\; a$
- Сравним b и $\mathrm{sqrt}(n)$
- $b \;=\; n \;/\; a \;>\; n \;/\; \mathrm{sqrt}(n)$ (очевидно)

Но все-таки вспомним основную тему занятия

И так, что же такое рекурсия?

И так, что же такое рекурсия?

- Рекурсия – когда функция вызывает саму же себя

И так, что же такое рекурсия?

- Рекурсия – когда функция вызывает саму же себя
- Примером может послужить даже математическая функция – функция Фибоначчи

И так, что же такое рекурсия?

- Рекурсия – когда функция вызывает саму же себя
- Примером может послужить даже математическая функция – функция Фибоначчи
- $F(n) = F(n - 1) + F(n - 2)$

И так, что же такое рекурсия?

- Рекурсия – когда функция вызывает саму же себя
- Примером может послужить даже математическая функция – функция Фибоначчи
- F(n) = F(n - 1) + F(n - 2)
- Один пример, который может быть вам поможет :

У попа была собака, он её любил, она съел кусок мяса, он её убил, в землю закопал, и надпись написал о том, что у попа была собака…..

Давайте может напишем свою рекурсивную функцию?

Это может очень глупо, но давайте попробуем реализовать факториал рекурсивно

Давайте подумаем над тем, как мы можем перебирать перестановки какие-либо

А теперь факт, который вас может шокировать: мы умеем возводить в степень за логарифм

А теперь факт, который вас может шокировать: мы умеем возводить в степень за логарифм

- Раньше мы могли возвести число k в степень n за n операций

А теперь факт, который вас может шокировать: мы умеем возводить в степень за логарифм

- Раньше мы могли возвести число k в степень n за n операций
- (k * k * k * …. * k)

А теперь факт, который вас может шокировать: мы умеем возводить в степень за логарифм

- Раньше мы могли возвести число k в степень n за n операций
- (k * k * k * …. * k)
- Давайте же теперь научимся делать это быстрее

А теперь факт, который вас может шокировать: мы умеем возводить в степень за логарифм

- Раньше мы могли возвести число k в степень n за n операций
- (k * k * k * …. * k)
- Давайте же теперь научимся делать это быстрее
- Для этого нам нужно несколько красивых фактов

Рассмотрим несколько ситуаций

n - чётно

n - чётно

- Тогда $k^n = k^{(n / 2)} * k^{(n / 2)} = k^{(n/2 + n/2)} = k^n$

n - чётно

- Тогда k^n = k^(n / 2) * k^(n / 2) = k^(n/2 + n/2) = k^n
- Это значит, что зная k^(n / 2) мы можем за одно действие вычислить k^n

n - нечётно

n - нечётно

- Тогда пусть nd = n / 2 (целочисленно)

n - нечётно

- Тогда пусть nd = n / 2 (целочисленно)
- k^n = k^nd * k^nd * k = k^(nd + nd + 1) = k^n

n - нечётно

- Тогда пусть nd = n / 2 (целочисленно)
- k^n = k^nd * k^nd * k = k^(nd + nd + 1) = k^n
- Но это можно облегчить, ведь k^n = k^(n - 1) * k, а в этом случае k (n - 1) – чётно, а с таким мы уже умеем работать.