

ЛЕКЦИЯ 07

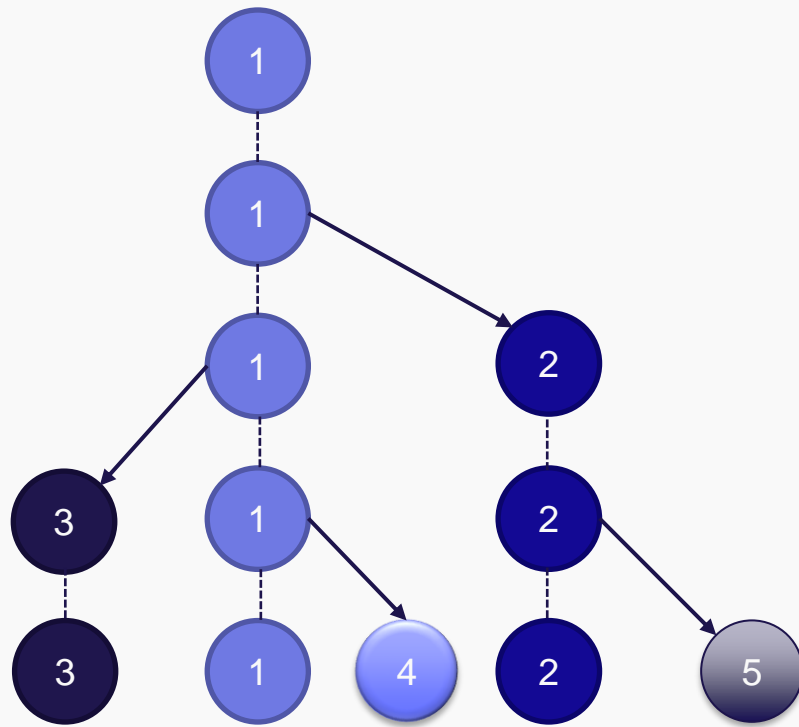
ЧИСЛА ФИБОНАЧЧИ

АЛГОРИТМИЗАЦИЯ И
ПРОГРАММИРОВАНИЕ



ЧИСЛА ФИБОНАЧЧИ

- Паре молодых кроликов нужен месяц, чтобы вырасти
- Каждая пара взрослых кроликов приносит пару молодых кроликов каждый месяц
- Сколько кроликов будет через год?



НАИВНЫЙ АЛГОРИТМ

$$F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5, F_6 = 8, F_7 = 13, \dots, F_n = F_{n-1} + F_{n-2}$$

Это ведет к очевидному решению:

```
unsigned long long fib(int n) {  
    if (n == 0) return 0ull;  
    if (n <= 2) return 1ull;  
    return fib(n - 1) + fib (n - 2);  
}
```

Вычислите на вашем компьютере `fib(10)`, `fib(20)`, `fib(50)`,
`fib(80)`

УЛУЧШИМ АЛГОРИТМ

Заменим рекурсию циклом и перепишем код.

```
unsigned long long fib(unsigned n) {  
    unsigned long long first = 0ull, second = 1ull; int idx;  
    if (n == 0) return 0ull;  
    for (idx = 2; idx <= n; ++idx) {  
        unsigned long long temp = second;  
        second += first;  
        first = temp;  
    }  
    return second;  
}
```

Вычислите на вашем компьютере `fib(10)`, `fib(20)`, `fib(50)`,
`fib(80)`

ФОРМУЛА ЧИСЕЛ ФИБОНАЧЧИ

Для вычисления n -ого числа Фибоначчи можно воспользоваться известной формулой:

$$F_n = \frac{\varphi_1^n - \varphi_2^n}{\sqrt{5}}, \text{ где } \varphi_1 = \frac{1 + \sqrt{5}}{2}, \varphi_2 = \frac{1 - \sqrt{5}}{2}$$

ФОРМУЛА ЧИСЕЛ ФИБОНАЧЧИ

Для вычисления n -ого числа Фибоначчи можно воспользоваться известной формулой:

$$F_n = \frac{\varphi_1^n - \varphi_2^n}{\sqrt{5}}, \text{ где } \varphi_1 = \frac{1 + \sqrt{5}}{2}, \varphi_2 = \frac{1 - \sqrt{5}}{2}$$

Для 80-ого числа Фибоначчи получим

`fib(80) = 23416728348467744`, хотя на самом деле

`fib(80) = 23416728348467685`

С чем связана ошибка? Насколько это плохо?

ПЕРЕПОЛНЕНИЯ

Вычислите на вашем компьютере `fib(91) ... fib(100)`

Что на экране?

ПЕРЕПОЛНЕНИЯ

Вычислите на вашем компьютере `fib(91)...``fib(100)`

Что на экране?

```
fib(91) = 4660046610375530309
```

```
fib(92) = 7540113804746346429
```

```
fib(93) = 12200160415121876738
```

```
fib(94) = 1293530146158671551 // oops!
```

До 93 числа все хорошо, но 94-е, очевидно, неверное, и должно быть 19740274219868223167

Давайте посчитаем эти числа в шестнадцатиричном виде

ПЕРЕПОЛНЕНИЯ

Вычислите на вашем компьютере `fib(91)...``fib(100)`

Что на экране?

```
fib(91) = 40abcfb3c0325745
```

```
fib(92) = 68a3dd8e61eccfbd
```

```
fib(93) = a94fad42221f2702
```

```
fib(94) = 11f38ad0840bf6bf
```

Видно, что число было срезано и должно быть равно `111f38ad0840bf6bf`

В компьютере выполняется не настоящее сложение, а сложение по модулю 2^{64}

РАЗРЯДЫ ЧИСЕЛ ФИБОНАЧЧИ

Числа Фибоначчи растут **очень** быстро

$\text{fib}(200) = 280571172992510140037611932413038677189525$

Увы, но это не влезет даже в 64 бита

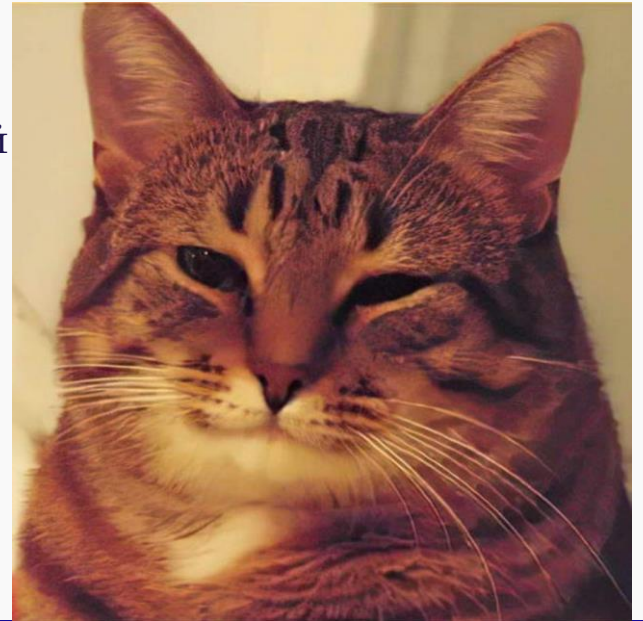


РАЗРЯДЫ ЧИСЕЛ ФИБОНАЧЧИ

Числа Фибоначчи растут **очень** быстро

$\text{fib}(200) = 280571172992510140037611932413038677189525$

Но мы можем использовать арифметику по модулю и подсчитать, например, последний разряд числа $\text{fib}(200) \% 10 = 5$



РАЗРЯДЫ ЧИСЕЛ ФИБОНАЧЧИ

Числа Фибоначчи растут **очень** быстро

$\text{fib}(200) = 280571172992510140037611932413038677189525$

Но мы можем использовать арифметику по модулю и подсчитать, например, последний разряд числа $\text{fib}(200) \% 10 = 5$

Последние разряды чисел Фибоначчи это числа Фибоначчи по модулю 10



РАЗРЯДЫ ЧИСЕЛ ФИБОНАЧЧИ

Числа Фибоначчи растут **очень** быстро

```
fib(200) = 280571172992510140037611932413038677189525
```

Но мы можем использовать арифметику по модулю и подсчитать, например, последний разряд числа $\text{fib}(200) \% 10 = 5$

Последние разряды чисел Фибоначчи это числа Фибоначчи **по модулю 10**

Напишите программу на C++, которая вычислит последний разряд числа $\text{fib}(331)$, используя улучшенный алгоритм и арифметику по модулю

УПРАЖНЕНИЕ

1. Обобщите улучшенный алгоритм нахождения n -ого числа Фибоначчи до функции нахождения n -ого числа Фибоначчи по любому модулю m
2. Напишите программу, которая выводит последовательности чисел Фибоначчи по модулям 2, 3, 4, 5, 6, 7, 8, 9, 10 хотя бы по 30 элементов для каждого модуля.
3. Какие вы видите закономерности?

ПЕРИОД ПИЗАНО

В результате выполнения предыдущего задания были замечены некоторые закономерности в последовательностях Фибоначчи по модулю m : они периодичны и новый период **всегда** начинается с 0, 1.

МАССИВЫ И МЕМОИЗАЦИЯ

```
unsigned pisanos[1000] = {0}; // означает {0, 0, ..., 0}
```

```
int fib_mod(unsigned long long n, unsigned m) {  
    assert(m > 0);  
    if (m < 1000) {  
        if (pisanos[m] == 0)  
            pisanos[m] = get_pisano_period(m);  
        n %= pisanos[m];  
    } else {  
        n %= get_pisano_period(m);  
    }  
}
```

Глобальный массив изначально инициализирован нулями, 0 маркирует невалидное значение

Мы можем частично инициализировать массив

МАССИВЫ И МЕМОИЗАЦИЯ

```
unsigned pisanos[1000] = {0, 1, 3, 8, 6, 20, 24, 16, 12};
```

```
int fib_mod(unsigned long long n, unsigned m) {  
    assert(m > 0);  
    if (m < 1000) {  
        if (pisanos[m] == 0)  
            pisanos[m] = get_pisano_period(m);  
        n %= pisanos[m];  
    } else {  
        n %= get_pisano_period(m);  
    }  
}
```

Все неинициализированные значения равны нулю

МАССИВЫ И МЕМОИЗАЦИЯ

Локальный массив по умолчанию не инициализирован и считывать его данные — серьезная ошибка

```
int foo() {  
    int wrong[100];  
    int correct[100] = {0};  
    std::cout << wrong[3]; // напечатает что угодно  
    std::cout << correct[3]; // напечатает 0  
    wrong[3] = 1;  
    std::cout << wrong[3]; // напечатает 1  
    return 0;  
}
```

Избегайте неинициализированных массивов и неинициализированных переменных в ваших программах.

МАССИВЫ И МЕМОИЗАЦИЯ

Массив — это поименованная область памяти фиксированного размера

СИСТЕМА СЧИСЛЕНИЯ ФИБОНАЧЧИ

Любое положительное число единственным образом представимо, как сумма Фибоначчи, если запретить в представлении две единицы рядом

$$8 = 10000_F = 5 + 3 = 1100_F = 5 + 2 + 1 = 1011_F$$

$$20 = 13 + 5 + 2 = 101010_F$$

$$30987 = 28657 + 1597 + 610 + 89 + 34$$

ДОМАШНЕЕ ЗАДАНИЕ

1. Обобщите улучшенный алгоритм нахождения n -ого числа Фибоначчи до функции нахождения n -ого числа Фибоначчи по любому модулю m
2. Напишите программу, которая выводит последовательности чисел Фибоначчи по модулям 2, 3, 4, 5, 6, 7, 8, 9, 10 хотя бы по 30 элементов для каждого модуля.
3. Напишите функцию, которая ищет длину периода Пизано
4. Обобщите улучшенный алгоритм нахождения n -ого числа Фибоначчи по любому модулю m для гигантских номеров и вычислите `fib(2816213588)` по модулю 30524.
5. Напишите программу, которая будет печатать число типа `unsigned int` в представлении системы чисел Фибоначчи. Мемоизируйте числа Фибоначчи, чтобы быстро искать ближайшее

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Дорохова Т.Ю., Основы алгоритмизации и программирования : учебное пособие для СПО / Т.Ю. Дорохова, И.Е. Ильина. – Саратов, Москва : Профобразование, Ай, Пи Ар Медиа, 2022. – 139 с.
2. Кудинов Ю.И., Основы алгоритмизации и программирования : учебное пособие для СПО / Ю.И. Кудинов, А.Ю. Келина. – 2-е изд. – Липецк, Саратов: Липецкий государственный технический университет, Профообразование, 2020. – 71 с.
3. Дональд Кнут, Искусство программирования. Том 1. Основные алгоритмы / Ю.В. Козаченко. - 3-е изд – Москва, Санкт-Петербург: ВИЛЬЯМС, 2018. – 721 с.