Введение: числа Фибоначчи

Александр Куликов

Онлайн-курс «Алгоритмы: теория и практика. Методы»
http://stepic.org/217

Определение

$$F_n = \begin{cases} 0, & n = 0\,, \\ 1, & n = 1\,, \\ F_{n-1} + F_{n-2}, & n > 1\,. \end{cases}$$

Определение

$$F_n = \begin{cases} 0, & n = 0\,, \\ 1, & n = 1\,, \\ F_{n-1} + F_{n-2}, & n > 1\,. \end{cases}$$

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

Скорость роста

Лемма

$$F_n \geq 2^{n/2} \text{ для } n \geq 6\,.$$

Скорость роста

Лемма

$$F_n \geq 2^{n/2} \text{ для } n \geq 6\,.$$

Доказательство

Индукция по n.

Скорость роста

Лемма

$$F_n \geq 2^{n/2} \text{ для } n \geq 6\,.$$

Доказательство

Индукция по n.

База: $F_6 = 8 = 2^{6/2}$, $F_7 = 13 > 2^{7/2} = 8\sqrt{2}$.

Скорость роста

Лемма

$$F_n \geq 2^{n/2} \text{ для } n \geq 6\,.$$

Доказательство

Индукция по n.

База: $F_6 = 8 = 2^{6/2}$, $F_7 = 13 > 2^{7/2} = 8\sqrt{2}$.

Переход: при $n \geq 8$

$$\begin{aligned} F_n = F_{n-1} + F_{n-2} &\geq 2^{(n-1)/2} + 2^{(n-2)/2} \\ &\geq 2 \cdot 2^{(n-2)/2} = 2^{n/2}. \qquad \square \end{aligned}$$

Формула

Факт

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right).$$

Формула

Факт

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right).$$

$$F_n \approx \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n = \frac{(1{,}618\ldots)^n}{\sqrt{5}}.$$

Экспоненциальная скорость роста

$$F_{20} = 6765$$

Экспоненциальная скорость роста

$$
\begin{aligned}
F_{20} &= 6765 \\
F_{50} &= 12586269025
\end{aligned}
$$

Экспоненциальная скорость роста

$$\begin{aligned} F_{20} &= 6765 \\ F_{50} &= 12586269025 \\ F_{100} &= 354224848179261915075 \end{aligned}$$

Экспоненциальная скорость роста

$$
\begin{aligned}
F_{20} &= 6765 \\
F_{50} &= 12586269025 \\
F_{100} &= 354224848179261915075 \\
F_{1000} &= 434665576869374564356885276750406258025646605173717804024817290895365554179490518904038798400792551692959225930803226347752096896232398733224711616429964409065331879382989696499285160037044761377951668492288 75
\end{aligned}
$$

Вычисление чисел Фибоначчи

Вычислить F_n

Вход: целое число $n \geq 0$.

Выход: F_n.

Вычисление чисел Фибоначчи

Вычислить F_n

Вход: целое число $n \geq 0$.

Выход: F_n.

Функция FibRecursive(n)

```
если n ≤ 1:
  вернуть n
иначе:
  вернуть FibRecursive(n − 1) + FibRecursive(n − 2)
```

Время работы

Через $T(n)$ обозначим число строк кода, выполняемых $\text{FibRecursive}(n)$.

Время работы

Через $T(n)$ обозначим число строк кода, выполняемых FIBRECURSIVE(n). Тогда

$$T(n) = 2 \text{ для } n \leq 1$$

и

$$T(n) = T(n-1) + T(n-2) + 3 \text{ для } n > 1\,.$$

Время работы

Через $T(n)$ обозначим число строк кода, выполняемых $\texttt{FibRecursive}(n)$. Тогда

$$T(n) = 2 \text{ для } n \leq 1$$

и

$$T(n) = T(n-1) + T(n-2) + 3 \text{ для } n > 1\,.$$

Следовательно, $T(n) \geq F_n$.

Время работы

Через $T(n)$ обозначим число строк кода, выполняемых $\textsc{FibRecursive}(n)$. Тогда

$$T(n) = 2 \text{ для } n \leq 1$$

и

$$T(n) = T(n-1) + T(n-2) + 3 \text{ для } n > 1\,.$$

Следовательно, $T(n) \geq F_n$.

$$T(100) \approx 1{,}77 \cdot 10^{21}$$

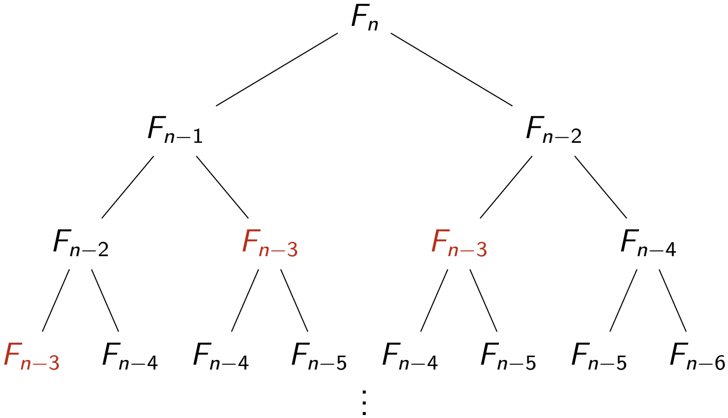
С частотой 1GHz будет работать несколько десятков тысяч лет.

Почему так медленно?

$$
\begin{tikzcd}[column sep=tiny]
& & & & & & & F_n \arrow[dllll, dash] \arrow[drrrr, dash] & & & & & & & \\
& & & F_{n-1} \arrow[dll, dash] \arrow[drr, dash] & & & & & & & & F_{n-2} \arrow[dll, dash] \arrow[drr, dash] & & & \\
& F_{n-2} \arrow[dl, dash] \arrow[dr, dash] & & & & F_{n-3} \arrow[dl, dash] \arrow[dr, dash] & & & & F_{n-3} \arrow[dl, dash] \arrow[dr, dash] & & & & F_{n-4} \arrow[dl, dash] \arrow[dr, dash] & \\
F_{n-3} & & F_{n-4} & & F_{n-4} & & F_{n-5} & & F_{n-4} & & F_{n-5} & & F_{n-5} & & F_{n-6}
\end{tikzcd}
$$

$$\vdots$$

Почему так медленно?



Другой алгоритм

Как бы мы вычисляли числа Фибоначчи вручную:
0, 1

Другой алгоритм

Как бы мы вычисляли числа Фибоначчи вручную:
0, 1, 1

$0 + 1 = 1$

Другой алгоритм

Как бы мы вычисляли числа Фибоначчи вручную:
0, 1, 1, 2

$0 + 1 = 1$
$1 + 1 = 2$

Другой алгоритм

Как бы мы вычисляли числа Фибоначчи вручную:
0, 1, 1, 2, 3

$0 + 1 = 1$
$1 + 1 = 2$
$1 + 2 = 3$

Другой алгоритм

Как бы мы вычисляли числа Фибоначчи вручную:
0, 1, 1, 2, 3, 5

$0 + 1 = 1$
$1 + 1 = 2$
$1 + 2 = 3$
$2 + 3 = 5$

Другой алгоритм

Как бы мы вычисляли числа Фибоначчи вручную:
0, 1, 1, 2, 3, 5, 8

$0 + 1 = 1$
$1 + 1 = 2$
$1 + 2 = 3$
$2 + 3 = 5$
$3 + 5 = 8$

Более быстрый алгоритм

Функция FibArray(n)

создать массив $F[0 \ldots n]$
$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
для i от 2 до n:
  $F[i] \leftarrow F[i-1] + F[i-2]$
вернуть $F[n]$

Более быстрый алгоритм

Функция $\textsc{FibArray}(n)$

создать массив $F[0 \dots n]$
$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
для i от 2 до n:
    $F[i] \leftarrow F[i-1] + F[i-2]$
вернуть $F[n]$

- Выполняет $2n + 2$ строки кода.
- $T(100) = 202$.

Заключение

- Ввели числа Фибоначчи.
- Наивный алгоритм работает непозволительно медленно даже на небольших входных данных.
- Улучшенный алгоритм работает очень быстро.

Секрет успеха — в правильном алгоритме.