

Динамическое программирование: числа Фибоначчи

Александр Куликов

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

Числа Фибоначчи

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, ...

Вычисление F_n

Вход: целое число $n \geq 0$.

Выход: F_n .

Функция `FIBRECURSIVE(n)`

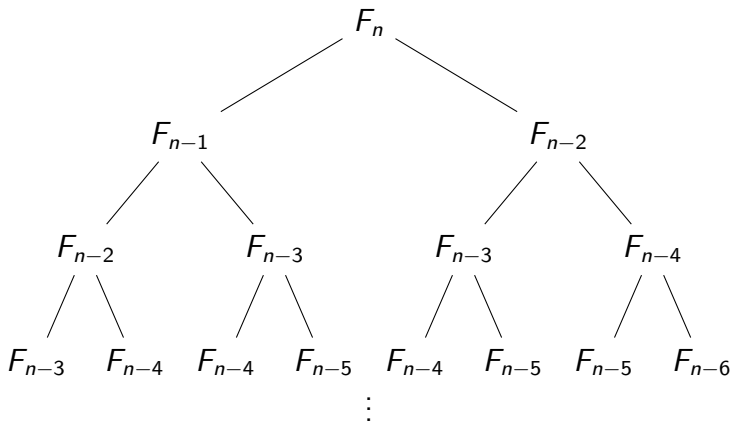
если $n \leq 1$:

 вернуть n

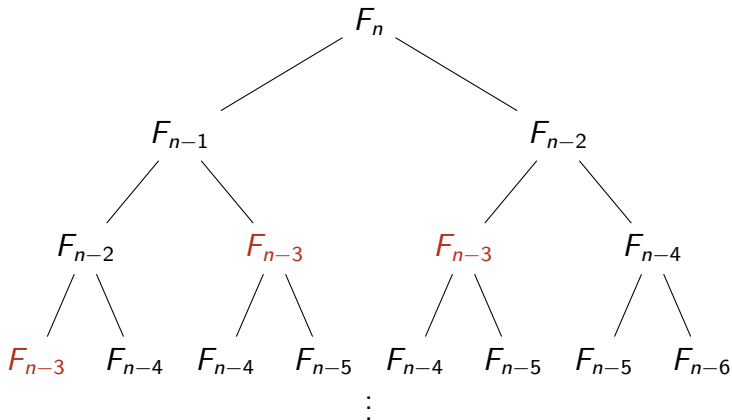
иначе:

 вернуть `FIBRECURSIVE($n - 1$) + FIBRECURSIVE($n - 2$)`

Дерево рекурсии



Дерево рекурсии



Дин. прог. назад

Инициализация

$$F[0 \dots n] = [-1, -1, \dots, -1]$$

Функция FIBTD(n)

если $F[n] = -1$:

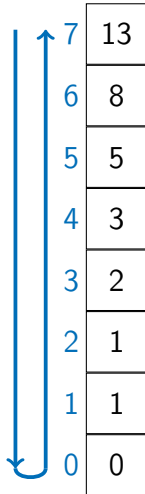
 если $n \leq 1$:

$$F[n] \leftarrow n$$

 иначе:

$$F[n] \leftarrow \text{FIBTD}(n-1) + \text{FIBTD}(n-2)$$

вернуть $F[n]$



7	13
6	8
5	5
4	3
3	2
2	1
1	1
0	0

Дин. прог. назад

Инициализация

$$F[0 \dots n] = [-1, -1, \dots, -1]$$

Функция $\text{FIBTD}(n)$

если $F[n] = -1$:

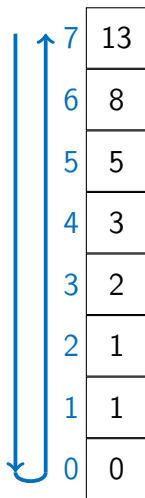
 если $n \leq 1$:

$$F[n] \leftarrow n$$

 иначе:

$$F[n] \leftarrow \text{FIBTD}(n-1) + \text{FIBTD}(n-2)$$

вернуть $F[n]$



7	13
6	8
5	5
4	3
3	2
2	1
1	1
0	0

Время работы: $O(n^2)$.

Дин. прог. вперёд

Функция FIBU(n)

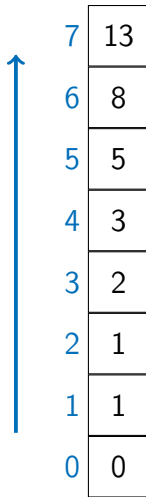
создать массив $F[0 \dots n]$

$F[0] \leftarrow 0, F[1] \leftarrow 1$

для i от 2 до n :

$F[i] \leftarrow F[i-1] + F[i-2]$

вернуть $F[n]$



7	13
6	8
5	5
4	3
3	2
2	1
1	1
0	0

Дин. прог. вперёд

Функция FIBU(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 $F[n]$

Время работы: $O(n^2)$.



7	13
6	8
5	5
4	3
3	2
2	1
1	1
0	0

Уменьшение памяти

Функция `FIBUIIMPROVED(n)`

если $n \leq 1$:

 вернуть n

$prev \leftarrow 0$

$curr \leftarrow 1$

повторить $(n - 1)$ раз:

$next \leftarrow prev + curr$

$prev \leftarrow curr$

$curr \leftarrow next$

вернуть $curr$

Заключение

- Вместо исходной задачи решается множество перекрывающихся подзадач. Ответы для подзадач хранятся в таблице.
- Динамическое программирование назад (или сверху вниз): рекурсивно от больших задач к меньшим.
- Динамическое программирование вперёд (или снизу вверх): итеративно от меньших задач к большим.
- Для некоторых задач можно уменьшить используемую память, проанализировав структуру таблицы.