

# Динамическое программирование

С. Лагутин 

2018 год

---

## Содержание

1. Динамическое программирование	3
2. Виды задач динамического программирования	4
3. Стандартные задачи динамического программирования	5
3.1. Числа Фибоначчи . . . . .	5
3.2. Задача о рюкзаке . . . . .	5

# 1. Динамическое программирование

*Динамическое программирование* — способ решения сложных задач путём разбиения на простые подзадачи.

Чтобы успешно решить задачу методом динамического программирования в общем случае нужно продумать пять пунктов:

1. Что храниться в качестве значения динамики и какие параметры однозначно определяют её состояние;
2. Начальные значения динамики;
3. Зависимости между состояниями — формула пересчёта значений динамики;
4. Порядок пересчёта;
5. Как вычисляется итоговый ответ — это могут быть какие-то значения посчитанной динамики или функция, использующая их.

Существует три основных порядка пересчёта:

1. Прямой порядок — состояние динамики пересчитывается из уже посчитанных;
2. Обратный порядок — из текущего состояния динамики обновляются зависящие от него;
3. «Ленивая динамика» — состояние пересчитывается рекурсивно с запоминанием значений динамики.

## 2. Виды задач динамического программирования

Задачи динамического программирования можно условно разделить по параметрам, которые требуются для определения значения динамики на следующие категории:

- Одномерная динамика;
- Многомерная динамика;
- Динамика по последовательностям;
- Динамика по подстрокам;
- Динамика по поддеревьям;
- Динамика по подмножествам;
- Динамика по профилю (прямоугольному и изломанному).

### 3. Стандартные задачи динамического программирования

- Числа Фибоначчи;
- Задача о рюкзаке;
- Поиск наибольшей общей подпоследовательности двух последовательностей;
- Поиск наибольшей возрастающей подпоследовательности;
- Кратчайший путь во взвешенном графе между двумя заданными вершинами (алгоритм Форда-Беллмана);
- Задача о коммивояжере — динамика по подмножествам;
- Максимальное независимое множество в дереве;
- Задача о замощении — динамика по профилю.

#### 3.1. Числа Фибоначчи

Вычисление  $n$ -ого числа Фибоначчи.

1.  $F_k$  — значение  $k$ -ого числа Фибоначчи;
2. Начальные значения:  $F_1 = F_2 = 1$ ;
3. Формула пересчёта:  $F_k = F_{k-1} + F_{k-2}$ ;
4. Порядок пересчёта — прямой;
5. Ответ:  $F_n$ .

#### 3.2. Задача о рюкзаке

Имеется  $N$  предметов,  $i$ -й предмет имеет массу  $w_i > 0$  и стоимость  $p_i > 0$ . Необходимо выбрать из этих предметов набор, имеющий суммарную массу не больше  $W$  (вместимость рюкзака), и при этом максимальную суммарную стоимость. В общем случае задача является NP-трудной. В данном варианте предполагается, что все  $W, w_i, p_i \in \mathbb{N}$ .

1.  $d[k, t]$  — максимальная стоимость из  $k$  первых предметов, которые имеют суммарную массу не больше  $w$ ;
2. Начальные значения:
  - $d[0, t] = 0, \forall t \in [0, W]$  — не берём ни одного предмета;
  - $d[k, 0] = 0, \forall k \in [0, N]$  — ни один предмет не поместится в рюкзак, вместимостью 0;

## 3. Пересчёт:

возможно две ситуации — либо берём  $k$  предмет, либо нет. Если не берём, то в рюкзаке будут предметы с номерами не больше  $k-1$ . Если предмет с номером  $k$  взять, то суммарная масса оставшихся предметов не превышает  $t - w_k$ , а общая стоимость увеличится на  $p_k$ . Необходимо выбрать максимальный из этих вариантов:

$$d[k, t] := \max(d[k-1, t], d[k-1, t - w_k] + p_k)$$

4. Порядок пересчёта: внешний цикл  $k = 1, \dots, N$ ;

внутренний цикл  $t = 0, \dots, W$ ;

5. Ответ:  $d[N, W]$ .