

ИНФОРМАТИКА



```
~/polytech $ date
```

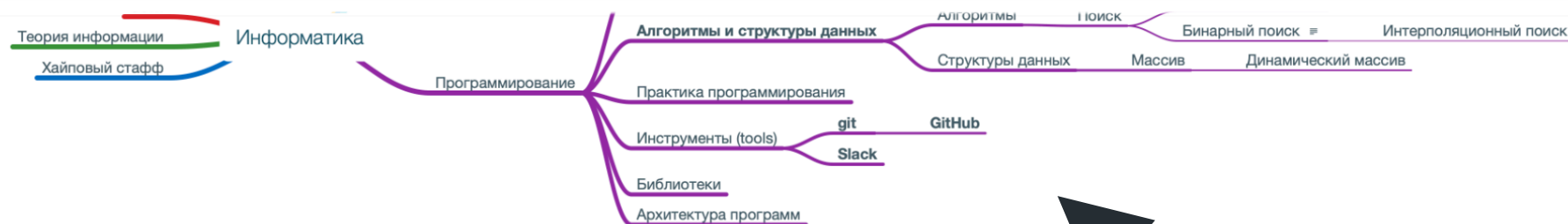
```
осень 2019
```

```
~/polytech $ name
```

```
константин кориков
```

```
~/polytech $ topic
```

```
проектирование алгоритмов 💀
```



Слайды (WIP)

1. Вводная лекция

- Как устроен курс.
- Инфраструктура: slack, github, docker, linux.
- Курс молодого питониста.
- Дополнительно: элементарная комбинаторика, решето Эратосфена, линейный конгруэнтный метод.

2. Алгоритмы и структуры данных. Начало

- Основные определения: алгоритм, АД, структура данных, тип данных.
- Нужная математика: числовые ряды, О-большое, рекуррентные формулы.
- Как сравнивать алгоритмы.
- Поиск: линейный, бинарный.
- Дополнительно: интерполяционный поиск, поиск прыжками, фибоначчиев поиск, экспоненциальный поиск.

3. Проектирование алгоритмов ☠

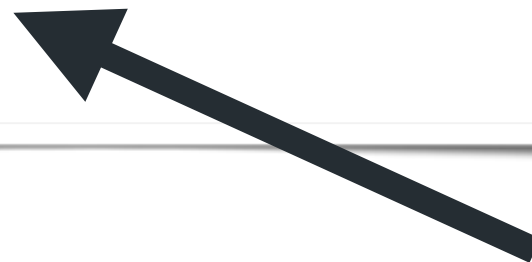
- Метод грубой силы (активное решение «в лоб»). Исчерпывающий перебор.

Mind map: Что уже прошли

Тема + короткое содержание

Сложная тема

- Обход дерева в глубину (DFS)
- Обход дерева в ширину (BFS)
- Очередь
- Очередь с приоритетом
- Пирамидальная сортировка (сортировка кучей)
- Поиск прыжками [см. [тема 2](#)]
- Префиксное дерево
- Решето Эратосфена [см. [тема 1](#)]
- Связный список



Словарик популярных алгоритмов и структур данных

Номер 2.1.1

2. Алгоритмы и структуры данных. Начало

1. Алгоритмы поиска

1. В системе регистрации сайта был сбой и в списке идентификаторов пользователей есть повторяющиеся идентификаторы. Напишите программу, которая проверяет есть ли повторы. Попробуйте найти решение за квазилинейное время.

►  Подсказка

2. Случаи ошибок участились, руководство просит ускорить систему поиска повторов из задачи №2.1.1. Попробуйте ускорить алгоритм до линейного времени.

►  Подсказка

Подсказка

О ЧЁМ ПОГОВОРИМ?

1. Метод «грубой силы»
2. Метод «преобразуй и властвуй»
3. Жадные методы
4. Метод «уменьшай и властвуй»
5. Метод «разделяй и властвуй»
6. Баланс память–время



«Грубая сила»

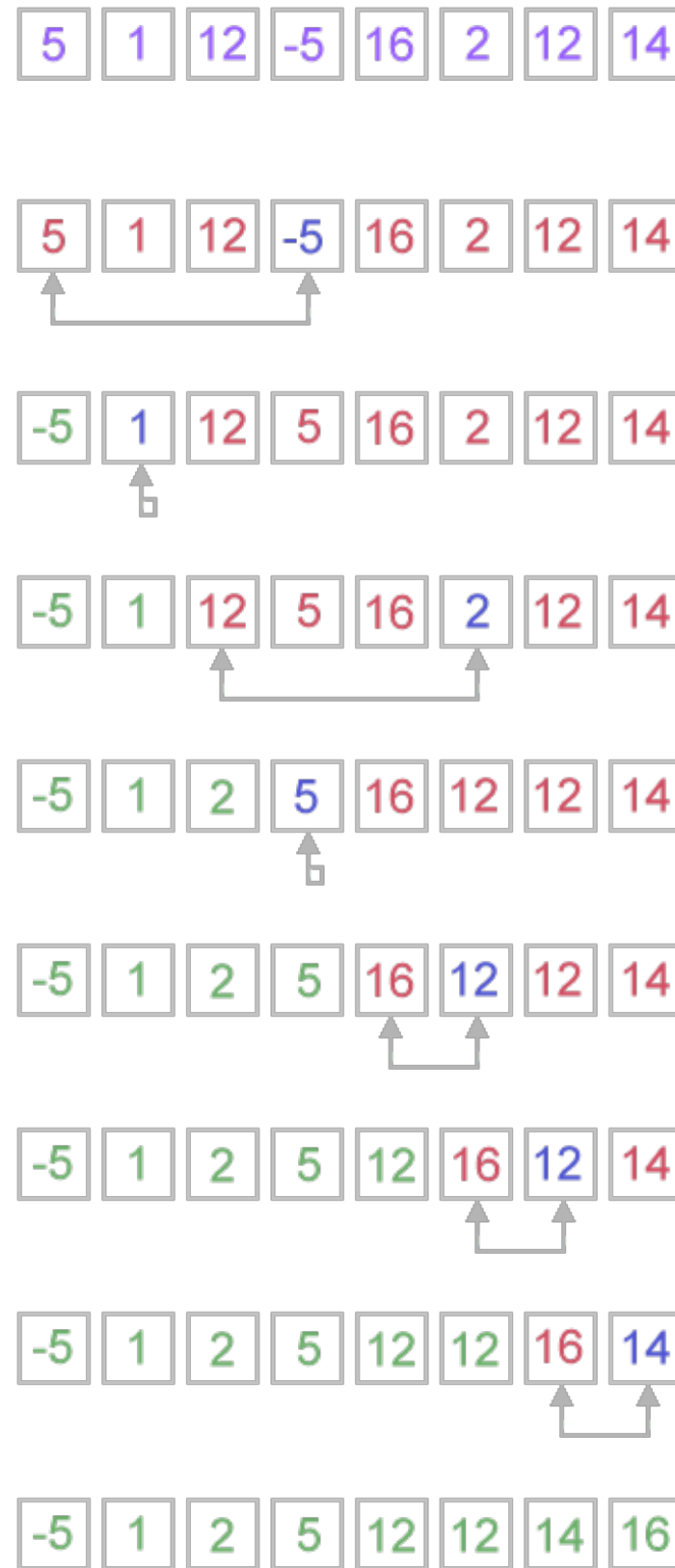
Метод «грубой силы»:

- обычно основан непосредственно на формулировке задачи,
- часто неэффективный в общем случае,
- дает хотя бы какое-то решение.



Сортировка

Сортировка
выбором





Задача о рюкзаке
(aka knapsack problem)



Что поместить?

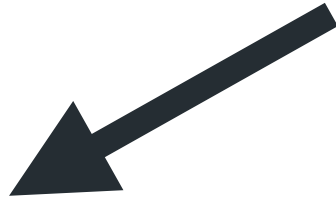


Исчерпывающий поиск (aka exhaustive search)

перебор всех вариантов

Комбинаторный взрыв!
NP-задача

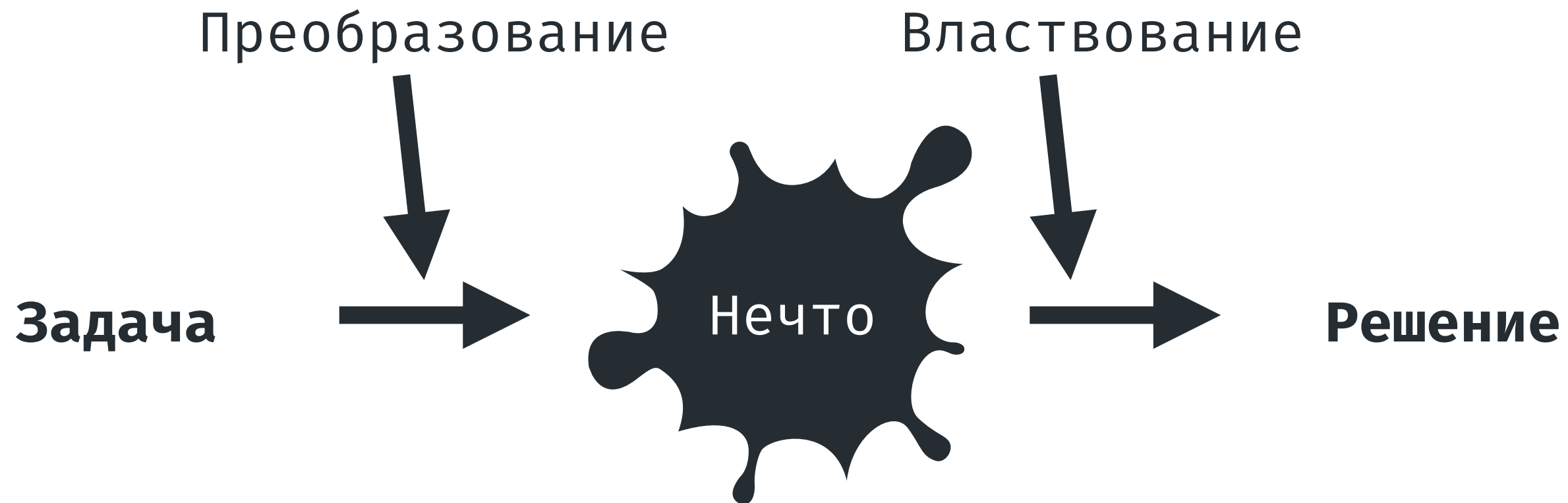
Отсекание заведомо
плохих веток перебора



Два метода:

- Backtracking
- Метод ветвей и границ

«Преобразуй и властвуй»



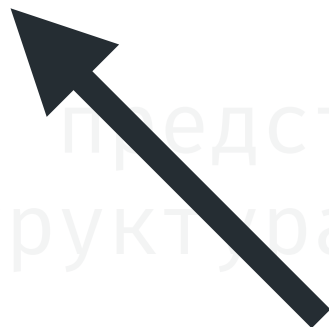
Как преобразовывать:

- Упрощать задачу
(структура данных не
меняется)
- Изменять представление
задачи (структура данных
меняется)
- Приводить к другой
задаче с известным
решением

Как преобразовывать:

- Упрощать задачу
(структура данных не
меняется)

- Изменять представление
задачи (структура данных
меняется)



Пример:

- Приводить к известной задаче с известным решением
предварительная сортировка

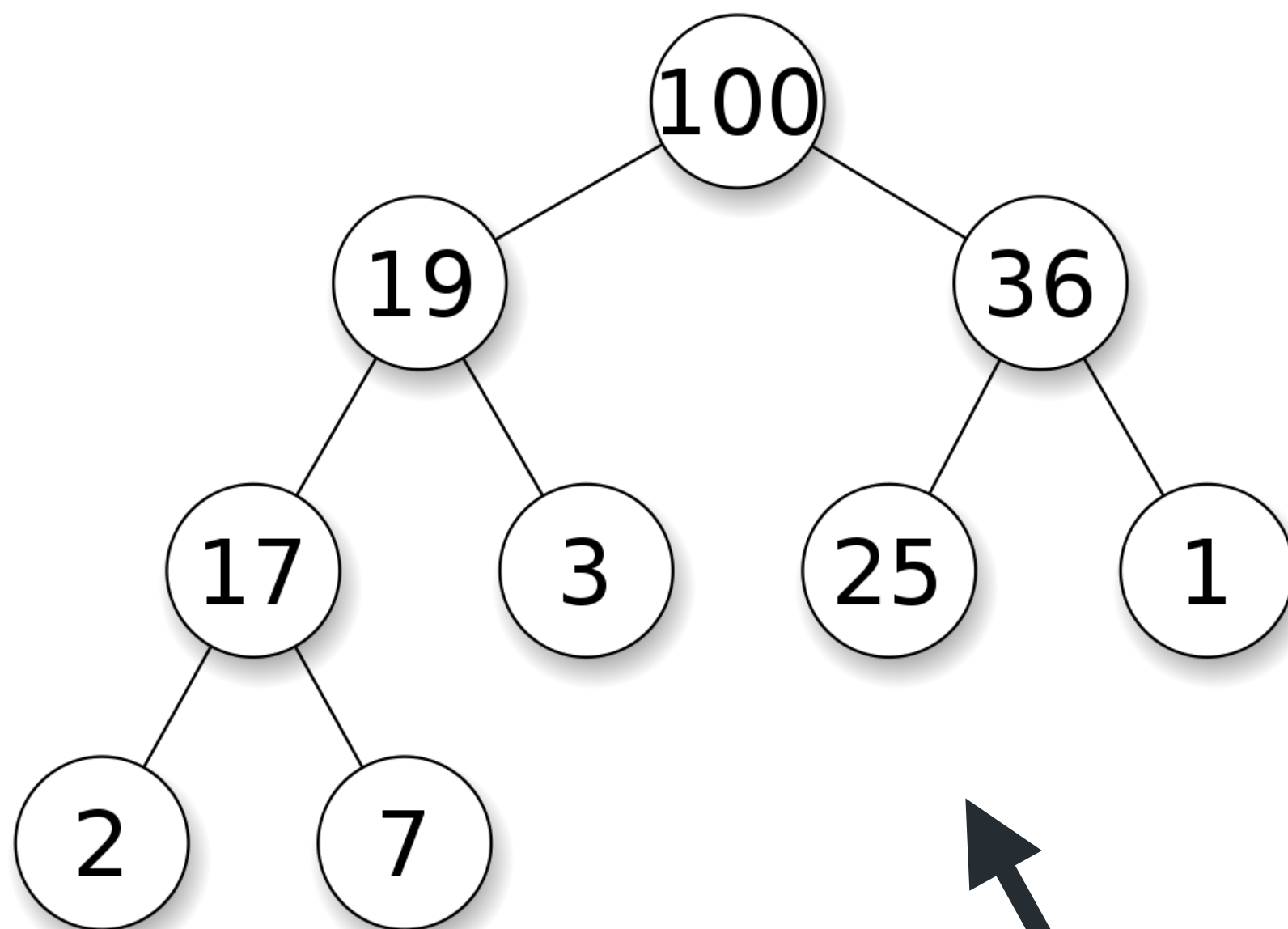
Как преобразовывать:

— Упрощать задачу
(структура данных не
меняется)

— Изменять представление
задачи (структура данных
меняется)

— Приводить к другой
задаче с известным

Пример: пирамидальная
сортировка
(ака сортировка кучей)



пирамида (ака куча)

Как преобразовывать:

— Упрощать задачу
(структура данных не
меняется)

Пример: максимизация
через минимизацию

— Изменять представление
задачи (структура данных
меняется)



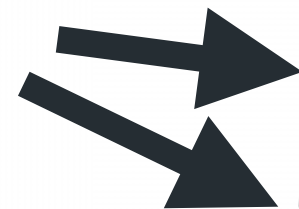
— Приводить к другой
задаче с известным
решением

Жадные методы

Метод математической индукции

способ доказательства
некоторого утверждения
 $P(n)$ для всех
натуральных n

Индукционный шаг
($n, n+1$)



База индукции
($n=1$)



Оптимизационная
задача

Надежный шаг

Решили?

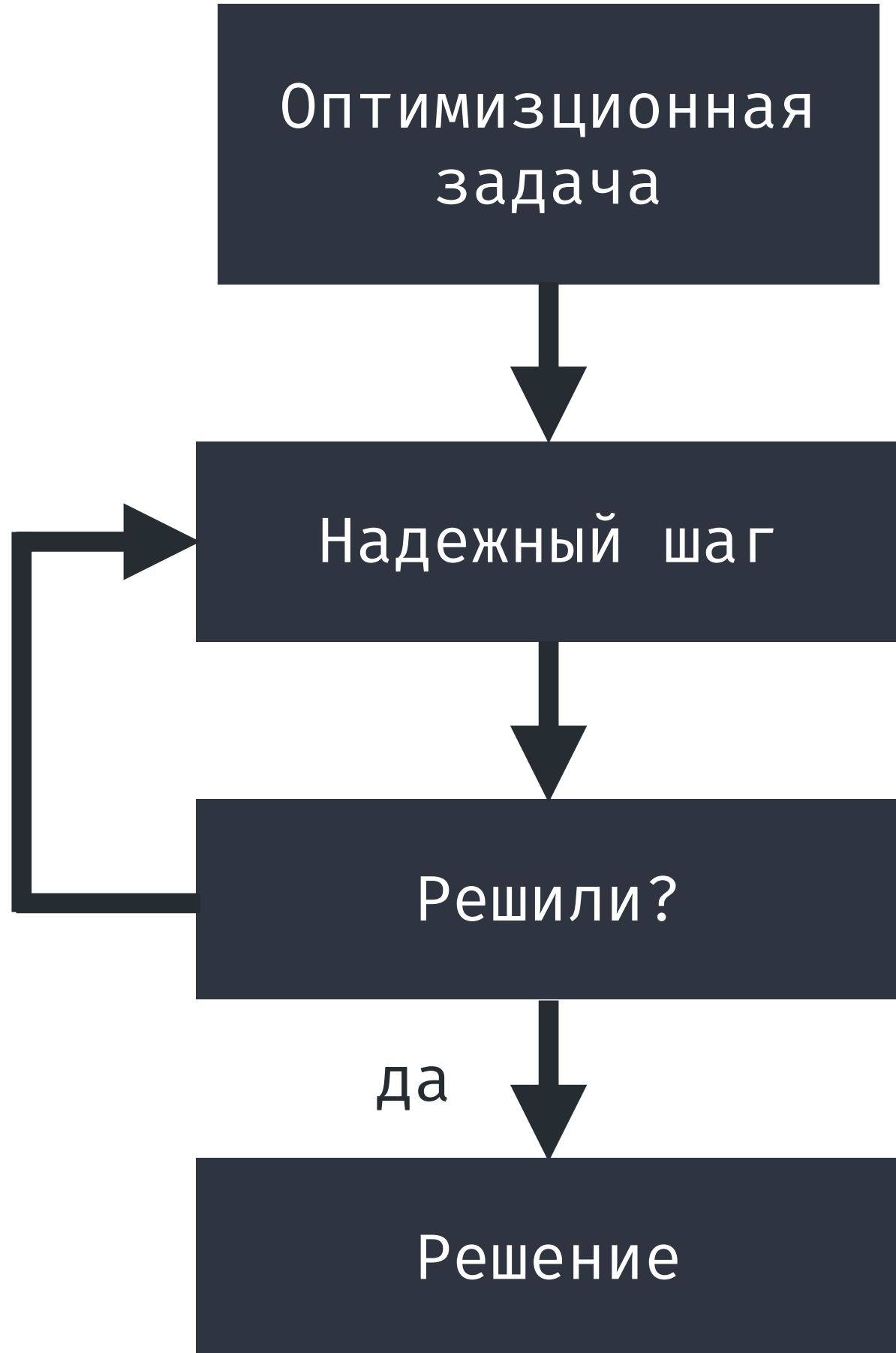
Решение

Оптимальное локально
Допустимое
Окончательное

Для некоторых задач
оптимальное глобально

нет

да

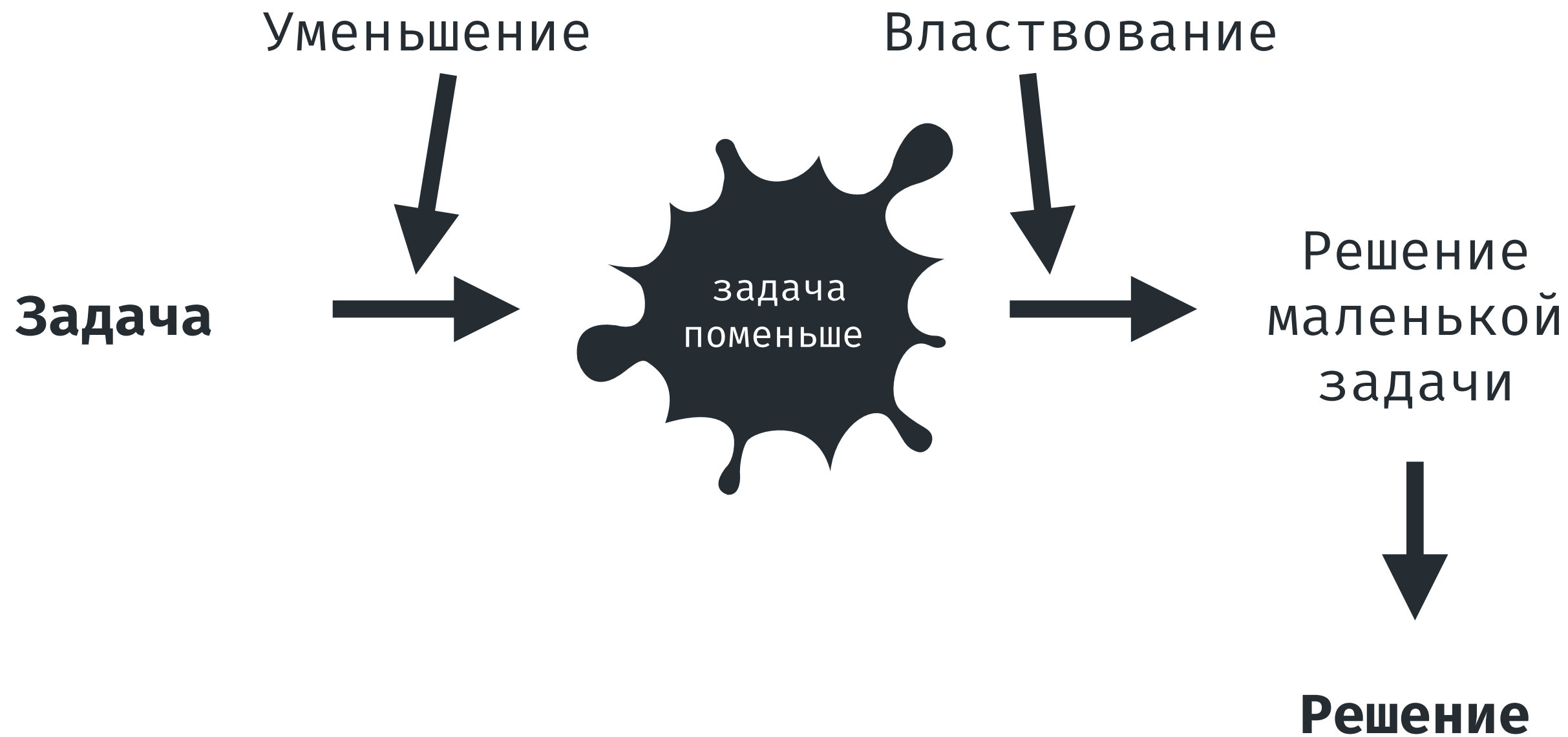




Что поместить?



«Уменьшай и властвуй»



Какие уменьшения бывают:

- на постоянную величину
- на постоянный множитель
- на переменную величину

Какие уменьшения бывают:

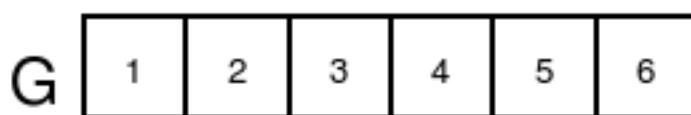
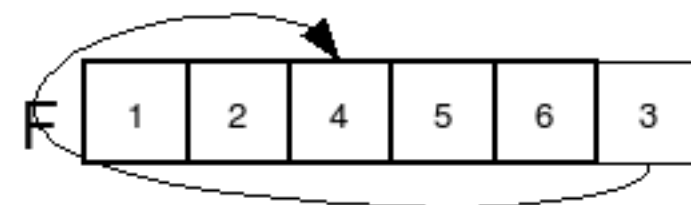
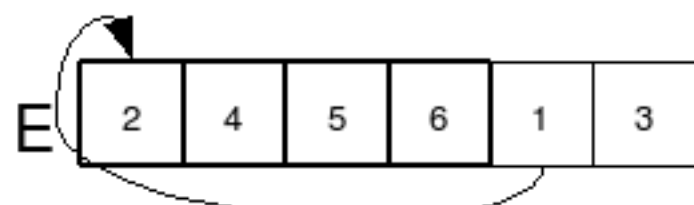
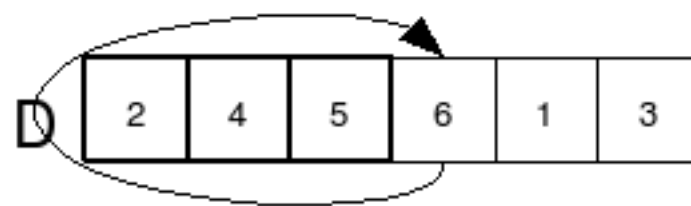
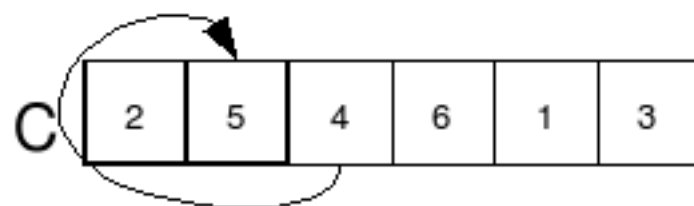
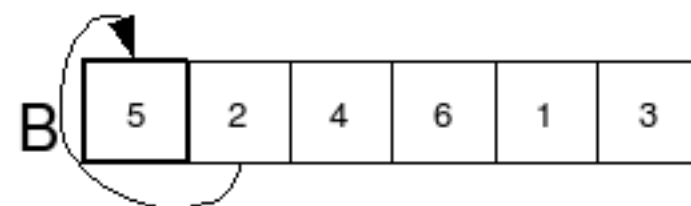
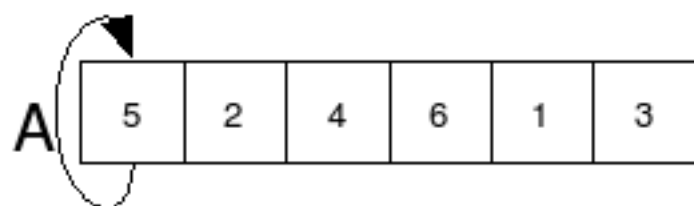
— на постоянную величину

— на постоянный множитель

— на переменную величину



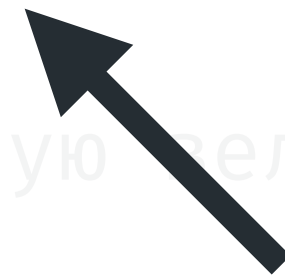
Пример:
сортировка
вставками



Уменьшаем
задачу

Какие уменьшения бывают:

- на постоянную величину
- на постоянный множитель
- на переменную величину

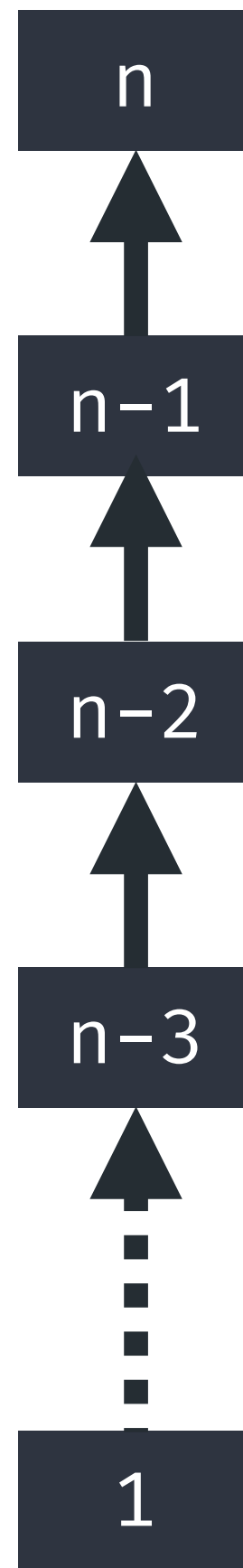
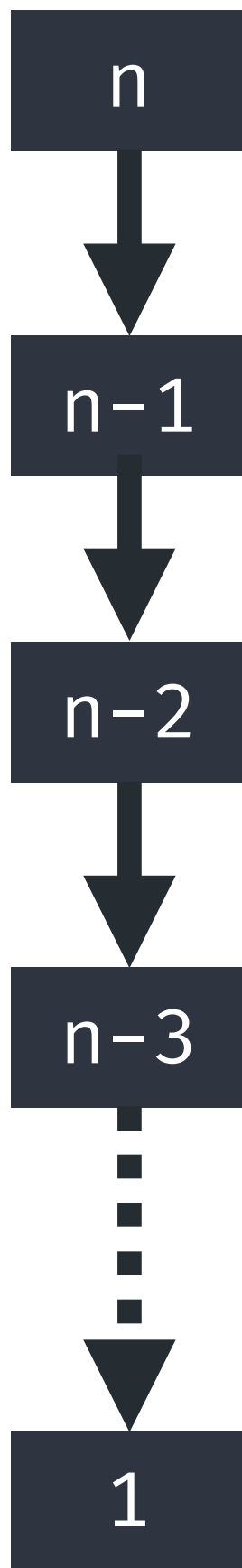


Пример: бинарный
поиск

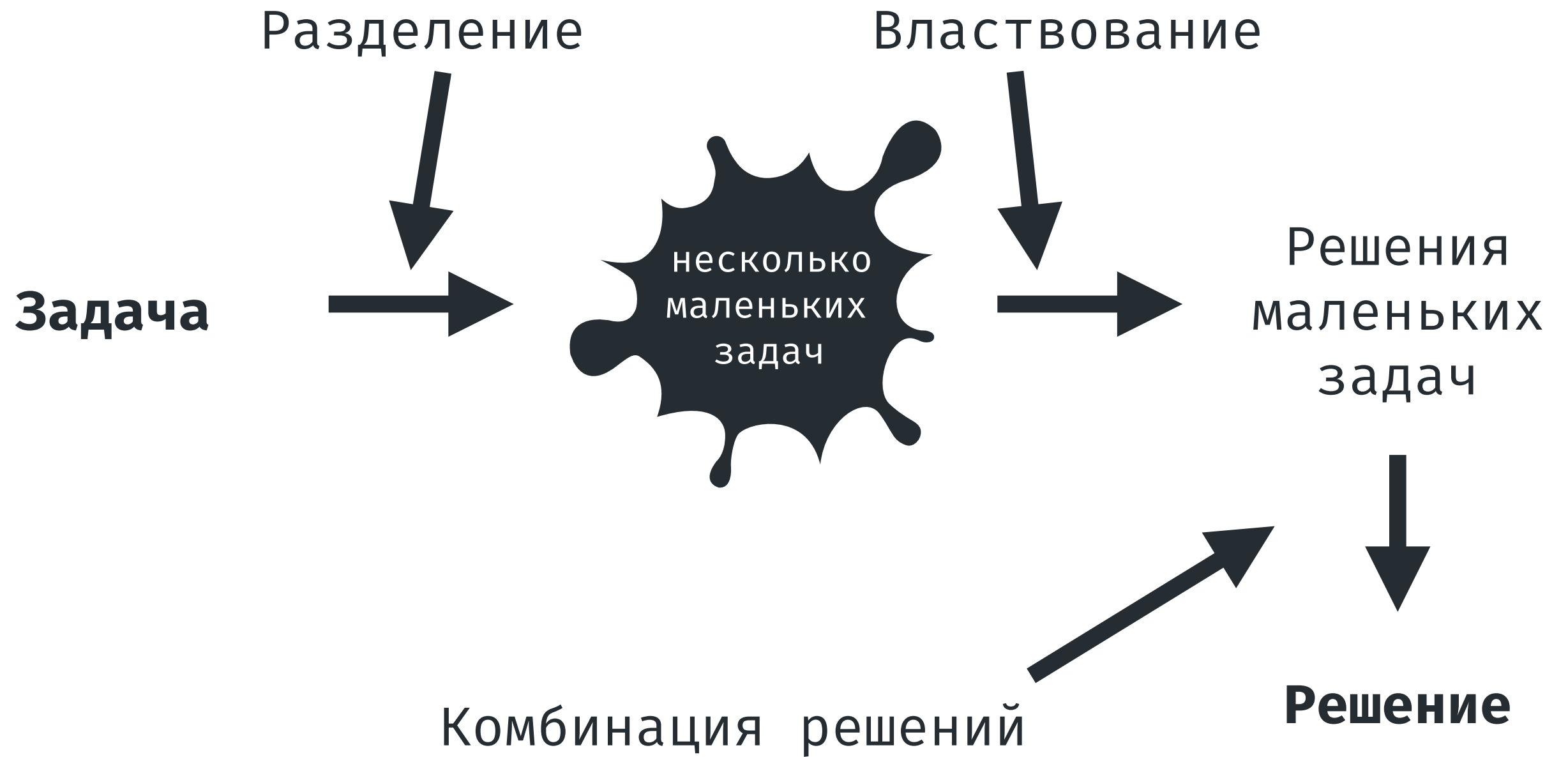
Какие уменьшения бывают:

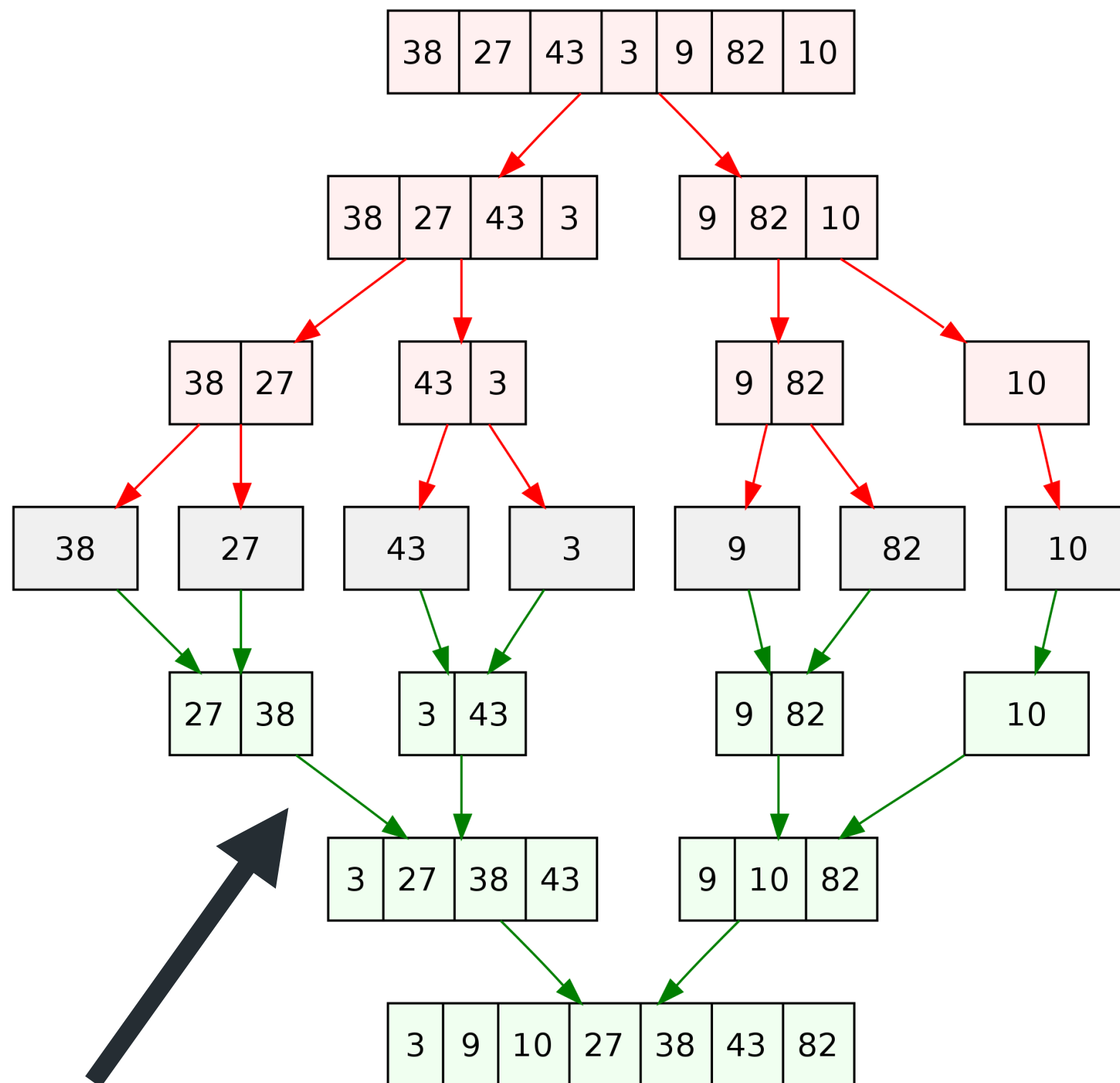
- на постоянную величину **Пример:**
- на постоянный множитель **интерполяционный поиск**
- на переменную величину

Рекурсивное
решение



«Разделяй и властвуй»





Сортировка
слиянием

Время работы

$$T(n) = aT(n/b) + f(n),$$

ОСНОВНАЯ ТЕОРЕМА. Если в рекуррентном уравнении $f(n) \in \Theta(n^d)$, где $d \geq 0$, то

$$T(n) \in \begin{cases} \Theta(n^d) & \text{если } a < b^d \\ \Theta(n^d \log n) & \text{если } a = b^d \\ \Theta(n^{\log_b a}) & \text{если } a > b^d \end{cases}$$

(Аналогичные результаты выполняются и для обозначений O и Ω .)



Основная теорема о рекуррентных соотношениях
(aka master theorem)

Тоже бывает сверху вниз
или снизу вверх

Баланс память–время

Лайфхак:

Если заранее посчитать и
запомнить ответ, то потом
можно как бы не считать

Три метода:

- улучшить входные данные
(дополняем данные информацией)
- предварительная
структуризация (данные не
трогаем, меняем доступ к ним)
- динамическое программирование

Три метода:

- улучшить входные данные
(дополняем данные информацией)

- предварительная
структуризация (данные не
трогаем, меняем доступ к ним)



Пример:

- динамическое программирование
 - сортировка
 - подсчет
 - распределения

Входная последовательность



	1	2	3	4	5	6	7	8
A	3	6	4	1	3	4	1	4

	1	2	3	4	5	6
C	2	0	2	3	0	1

(a)

	1	2	3	4	5	6
C	2	2	4	7	7	8

(b)

	1	2	3	4	5	6	7	8
B							4	

	1	2	3	4	5	6
C	2	2	4	6	7	8

(c)

	1	2	3	4	5	6	7	8
B		1					4	

	1	2	3	4	5	6
C	1	2	4	6	7	8

(d)

	1	2	3	4	5	6	7	8
B		1				4	4	

	1	2	3	4	5	6
C	1	2	4	5	7	8

(e)

	1	2	3	4	5	6	7	8
B	1	1	3	3	4	4	4	6

(f)



Выходная
последовательность

Три метода:

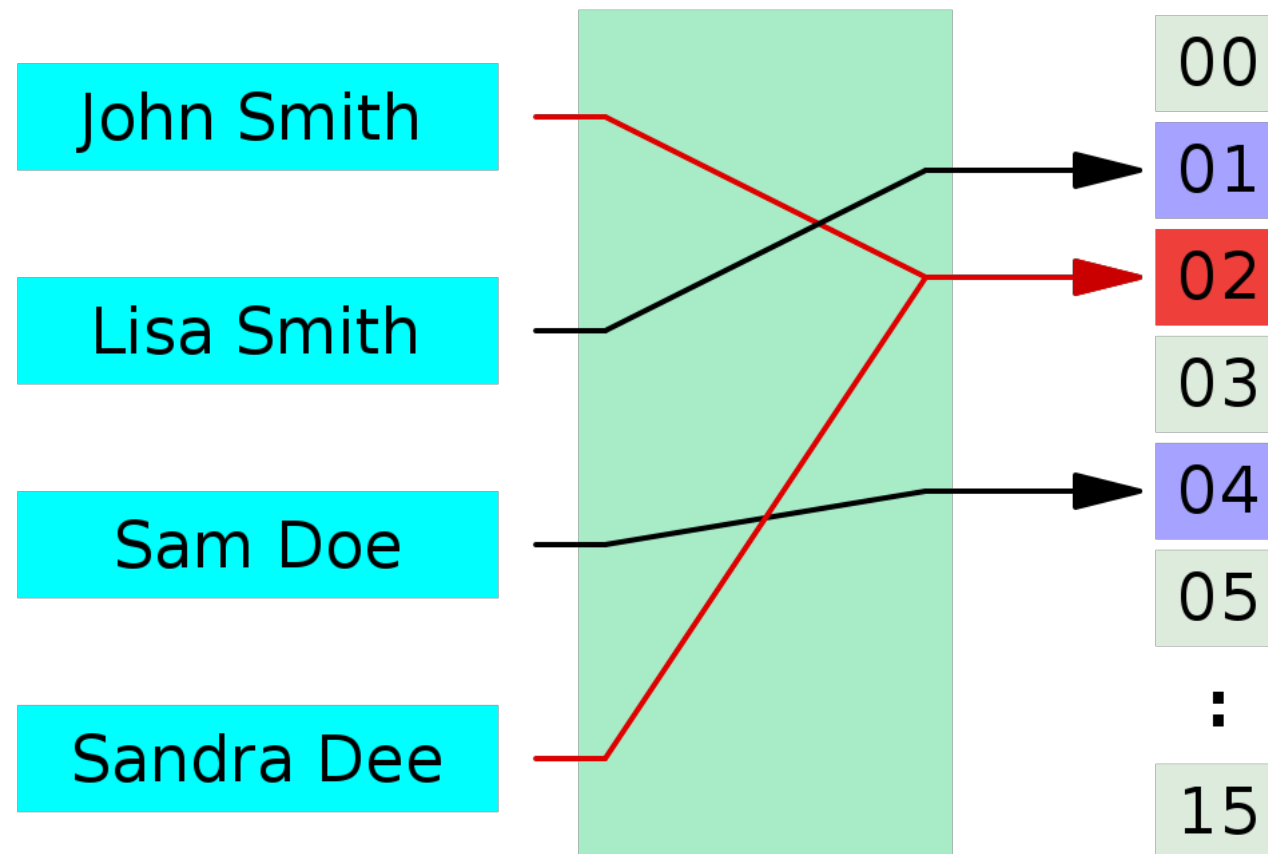
Пример:
хеширование

- улучшить входные данные
(дополняем данные информацией)
- предварительная
структуризация (данные не
трогаем, меняем доступ к ним)
- динамическое программирование

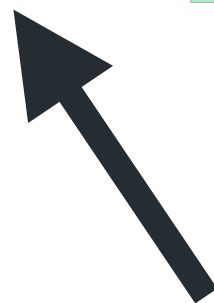
хеш (aka hash)



хеш-функция



КЛЮЧ



Три метода:

— улучшить входные данные
(дополняем данные информацией)

Когда подзадачи

— предварительная перекрываются
структуризация (данные не
трогаем, меняем доступ к ним)



— динамическое программирование

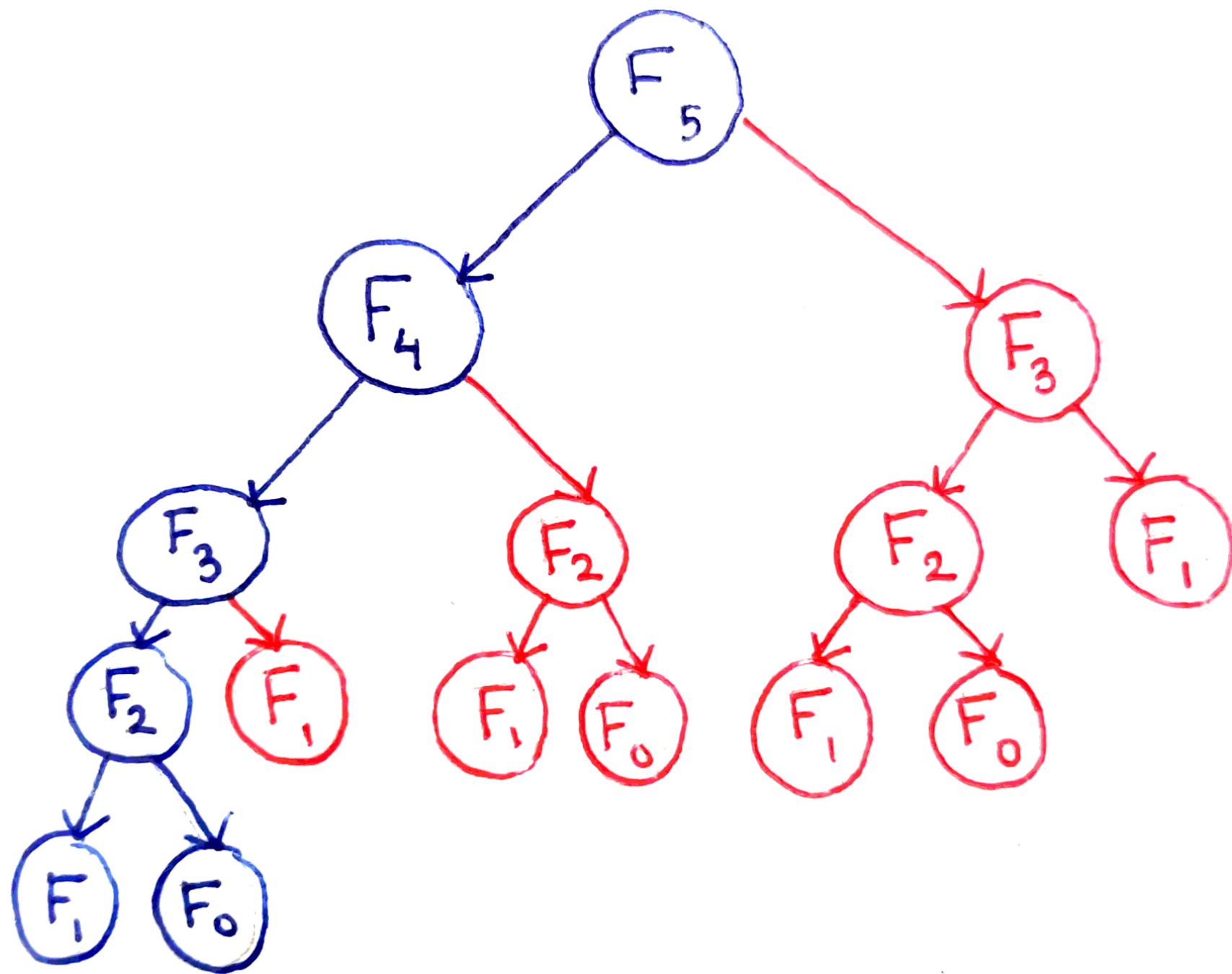
С помощью мемоизации



Тоже бывает **сверху вниз**
или **снизу вверх**



С помощью табуляции



Алгоритмы *Введение в* разработку и анализ



Ананий Левитин

