

# Разработка эффективных алгоритмов

# Методы разработки эффективных алгоритмов

1. Жадные алгоритмы
2. Рандомизированные алгоритмы
3. Метод «Разделяй и властвуй»
4. Динамическое программирование

# Жадные алгоритмы

1. Задача о процессах
2. Задача о рюкзаке
3. Коды Хаффмана ( алгоритм сжатия)

# Суть жадных алгоритмов

Всегда делается выбор, который является лучшим в данный момент в надежде, что он приведет к оптимальному решению

# Рандомизированные алгоритмы

- Задача о найме
- Парадокс дней рождений
- Шары и корзины
- Орел или решка

# Суть ранд. алгоритмов

Эффективность алгоритма рассматривается не для худшего случая, а для тех случаев, которые встречаются чаще всего.

# Задача о найме

Предприниматель(П) хочет найти нового менеджера. Агентство(А) будет присылать по 1 кандидату в день. П проводит собеседование и принимает решение о найме. За каждого кандидата П платит А некоторую сумму. Задача П – нанять самого достойного. Как только приходит лучший, то он заменяет текущего менеджера.

Определить расходы П.

# Эффективность

В худшем случае –  $O(cn)$

В среднем случае –  $O(c \ln n)$

$c$  – стоимость найма

$n$  – количество сотрудников



# Парадокс дней рождений

Сколько людей нужно собрать в одной комнате, чтобы вероятность совпадения даты рождения у 2 из них достигла 50%.

# Парадокс дней рождений

Ответ: 23

Количество людей необходимо  $\sim \Theta(n^{1/2})$

$n$  – количество дней в году

# Шары и корзины

Имеется  $n$  корзин. Независимо друг от друга в каждую опускаются шары. Сколько шаров понадобится, чтобы в каждой корзине оказался шар?

# Шары и корзины

Ответ:  $n \ln n$

# Задача сборщика купонов

Сколько нужно купить купонов, чтобы собрать всю коллекцию из  $n$  штук?

Примерно  $n \ln n$

# Орел или решка

Монета подбрасывается  $n$  раз. Сколько раз наверняка выпадет орел?

# Орел или решка

Ответ:  $\Theta(\ln n)$ .

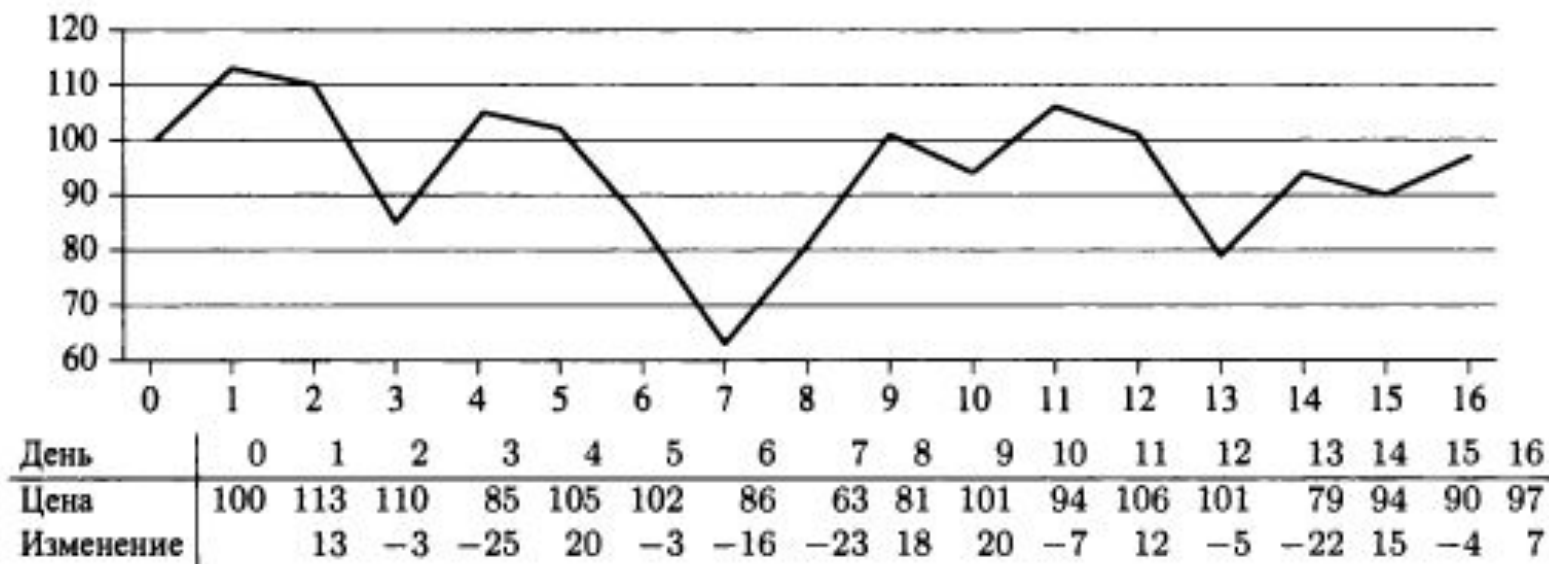
# Метод «разделяй и властвуй»

- 1. Разделение** – деление на несколько задач, которые являются меньшими экземплярами той же задачи
- 2. Властвование** – решение подзадач с помощью рекурсии
- 3. Комбинирование** – решение подзадач в решении исходной задачи.



# Задача о акциях

Имеется график роста цены акций



# Задача о акциях

Можно купить акции в 1 день и через некоторый момент их продать. (1 раз)

Задача – получить максимальную прибыль.

# Задача о акциях

## 1 Способ – Перебор

Выбрать все варианты с выбором начальной и конечной дат. Получим

$$T(n) = \Omega(n^2)$$

$n$  – количество дней.

# Задача о акциях

## 2 Способ – «Разделяй и властвуй»

Суть – необходимо найти последовательной дней, для которых игровая разница будет максимальной, т.е. нужно решить задачу о **максимальном**

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	3	-25	20	-3	-16	-23	18	20	-7	12	-5	-22	15	-4	7

# Разделение

Разобьем на массив на 2 равные части

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	3	-25	20	-3	-16	-23	18	20	-7	12	-5	-22	15	-4	7

# Разделение

Варианты:

1. Максимальный массив находится в 1 части.
2. Максимальный массив находится в 2 части.
3. Середина исходного массива находится внутри максимального массива

# Властвование

1 и 2 Варианты являются меньшими экземплярами исходной задачи, значит для них применим рекурсивный алгоритм

# Комбинирование

3 вариант сводится к нахождению максимального массива состоящего из 2 максимальных массивов:

- с концом в середине исходного массива

- началом в середине исходного

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	3	-25	20	-3	-16	-23	18	20	-7	12	-5	-22	15	-4	7



# Сложность алгоритма

$T(n)$  – исходная сложность

$T(n/2)$  – сложность 1 и 2 варианта

$\Theta(n)$  – сложность 3 варианта

$$T(n) = 2T(n/2) + \Theta(n)$$

$$T(n) = \Theta(n \lg n)$$

# Метод «разделяй и властвуй»

Сортировка массива

1. Сортировка слиянием с балансировкой

$$T(n) = 2T(n/2) + Cn$$

2. Сортировка выбором

$$T(n) = T(n-1) + Cn$$

# Метод «разделяй и властвуй»

Поиск максимального и минимального элементов:

1. Выбираем  $\min$  и  $\max$  как 1 элемент и сравниваем со всеми остальными

$$T(n) = 2n - 3$$

2. Делим массив на 2 части с балансировкой

и находим  $\max$  и  $\min$  в каждой части

$$T(n) = 3n/2 + 2$$

# Динамическое программирование

1. Задача погружается в семейство задач той же природы. Другими словами, разбивается на зависимые (могут пересекаться) задачи.
2. Каждая подзадача решается отдельно один раз. Оптимальные значения решений всех подзадач запоминаются (обычно в таблице).
3. Для исходной задачи строится возвратное соотношение, связывающее между собой оптимальные значения зависимых

# Задача о ступеньках

Дети играют в игру, в ходе которой игроку нужно подняться по лестнице, на каждой из  $N$  ступенек которой лежит некоторое число штрафных фишек  $A_i$  ( $i$  от 1 до  $N$ ). Если игрок наступает на ступеньку, он забирает себе все штрафные фишки с этой ступеньки. С любой ступеньки игрок может шагнуть на следующую ступеньку или перешагнуть одну ступеньку. Для каждой ступеньки известно число штрафных фишек, лежащих на ней. Найти такую последовательность ступеней, наступая на которые, игрок соберёт минимальное число штрафных фишек.

# Задача о ступеньках

1. Если мы стоим на  $k$  ступеньке, то на нее мы можем попасть только из  $k-1$  и  $k-2$ .
2. Пусть  $S(k)$  – количество штрафных фишек  
Тогда  
$$S(k) = \min(S(k-1)+A_k, S(k-2)+A_k).$$

# Задача о ступеньках

Массив A

1	2	3	4	5	6
1	4	5	2	3	6

Массив S

1	2	3	4	5	6	7
1	4	6	6	9	12	9

Ответ: 1, 3, 5 или 2, 4, 5