



# Алгоритмы и структуры данных

Лекция 3\*. Задачи на построение алгоритмов.

(с) Глухих Михаил Игоревич, [glukhikh@mail.ru](mailto:glukhikh@mail.ru)

## Аттестация 2 из 3: соревнование

- 1 октября в 16:00 в 9-325 (вместо лекции) Михаил Анатольевич Беляев расскажет подробности и ответит на вопросы

# Аттестация 2 из 3: упражнения

- ▶ Решения задач из учебного проекта

## Учебный проект

- <https://github.com/Kotlin-Polytech/Algorithms-2018>
- Интегрирован в систему Котоед (Algorithms 2018)
- Примеры к лекциям + задачи
- 6 уроков (1-3, 5-7)
- Задачи можно решать на Kotlin или Java (есть отдельные задачи только на Kotlin)

# Аттестация 2 из 3: упражнения

- Решения задач из учебного проекта
- Состав уроков
  1. Задачи на сортировку
  2. Общие задачи на построение алгоритмов
  3. Задачи на деревья
  4. Пока отсутствует
  5. Задачи на графы
  6. Задачи на динамическое программирование
  7. Задачи на эвристические алгоритмы

## Аттестация 2 из 3: упражнения

- Решения задач из учебного проекта
- Требования к количеству
  - Минимум: по одной задаче из трёх уроков
  - Норма: по две задачи из четырёх уроков
  - Оптимум: по две задачи из пяти-шести уроков

# Аттестация 2 из 3: упражнения

- ▶ Решения задач из учебного проекта
- ▶ Требования к решению
  - ▶ Код (в приличном стиле)
  - ▶ В комментарии (обязательно)
    - ▶ Оценка трудоёмкости
    - ▶ Оценка ресурсоёмкости – можно опустить, только если  $O(1)$
  - ▶ Тесты **каждой** из следующих групп:
    - ▶ Обычные случаи
    - ▶ Краевые случаи (мало данных, нестандартный ответ и т.п.)
    - ▶ Длинные (на производительность)
    - ▶ В некоторых задачах могут уже быть тесты всех групп (**не во всех**)
    - ▶ **Обязательно** дописать к тестам задачи **хотя бы одну** проверку
  - ▶ Тесты должны проходить

## Аттестация 2 из 3: упражнения

- Решения задач из учебного проекта
- Требования к датам
  - 10 декабря 16:00 – момент, когда приём задач **заканчивается**
    - 3 декабря – рассылка с текущим положением дел
  - 5 ноября 16:00 – заканчивается приём задач уроков 1-3, далее принимаются только уроки 5-7
    - 29 октября – рассылка с текущим положением дел



# Аттестация 2 из 3: упражнения

- Решения задач из учебного проекта
- Процесс
  - Всё (или почти всё) проверяю я
  - Делать по десять итераций мы не будем (одна-две это максимум)
  - В последнюю неделю перед дедлайном замечаний для исправления уже не будет

# Аттестация 2 из 3: упражнения

- Решения задач из учебного проекта
- От чего зависит оценка
  - Количество и сложность задач
    - Из каждого урока оцениваются две самые сложные задачи
  - Качество кода + Качество алгоритма
  - Правильность оценки алгоритма
  - Полнота тестов

# Максимальный подмассив

- Есть (известен) массив цен на акции `Price[]`
- Требуется выбрать два дня  $i, j$ , такие, что
  - $j > i$  (!!!)
  - $\text{Price}[j] - \text{Price}[i] \rightarrow \text{MAX}$
- То есть ВНАЧАЛЕ купить дешево, ПОТОМ продать дорого
- Примеры
  - 100, 113, 110, 85, 105, 102, 86, 63, 81, 101, 94, 106, 101, 79, 94
  - 10, 11, 7, 10, 6

# Максимальный подмассив

- Есть (известен) массив цен на акции `Price[]`
- Требуется выбрать два дня  $i, j$ , такие, что
  - $j > i$  (!!!)
  - $\text{Price}[j] - \text{Price}[i] \rightarrow \text{MAX}$
- То есть ВНАЧАЛЕ купить дёшево, ПОТОМ продать дорого
- Примеры
  - 100, 113, 110, 85, 105, 102, 86, 63, 81, 101, 94, 106, 101, 79, 94
  - 10, 11, 7, 10, 6

# Вариация

- Берём массив цен  $Price[]$
- Формируем массив изменений  $Delta[]$  :  
 $Delta[i] = Price[i+1] - Price[i]$
- Требуется найти подмассив  $Delta$ , такой, что сумма его элементов максимальна
  - МАКСИМАЛЬНЫЙ ПОДМАССИВ

# Варианты решения

- В лоб
- Разделяй и властвуй
- Рекуррентный

# Варианты решения

- В лоб
  - Просто перебираем все варианты:  $O(N^2)$
- Разделяй и властвуй
  - Разбиваем массив дельт на две половины
  - В каждой ищем максимальный подмассив
  - Сравниваем его с тем, который проходит через обе половины

# Варианты решения

- Разделяй и властвуй
  - Разбиваем массив дельт на две половины
  - В каждой ищем максимальный подмассив
  - Сравниваем его с тем, который проходит через обе половины
- Рекуррентный
  - Ищем максимальный подмассив для массива размером  $J-1$
  - Далее одно из двух
    - Либо он максимален и для размера  $J$
    - Либо для размера  $J$  максимален подмассив, включающий последний элемент



# Задача умножения матриц

- ▶ Упрощённый вариант: квадратные матрицы размером  $M \times M$ , причём  $M$  является степенью двойки
- ▶ Вариант “в лоб”
  - ▶ Вычисление каждого элемента произведения требует  $M$  умножений и  $M-1$  сложений
  - ▶ Элементов имеется  $M^2$
  - ▶ Общая трудоёмкость  $O(M^3)$
- ▶ Рекурсивный вариант
  - ▶ Делим каждую матрицу на четыре подматрицы
    - ▶  $A_{11} \ A_{12} \ B_{11} \ B_{12}$
    - ▶  $A_{21} \ A_{22} \ B_{21} \ B_{22}$
  - ▶ Перемножаем подматрицы отдельно, потом сливаем вместе
  - ▶  $S(M) = 8S(M/2) + O(M^2) \implies S(M) = O(M^3)$

# Метод Штрассена

- Идея: уменьшить число перемножений подматриц с 8 до 7, сведя рекуррентное соотношение к следующему
  - $S(M) = 7S(M/2) + O(M^2) \implies S(M) = O(M^{2.81})$
- С этой целью из 8 подматриц  $A_{ij} / B_{ij}$  формируется ещё 10 матриц  $S_k = A_{ij} +/- B_{mn}$ , вычисляется 7 произведений, суммируются результаты для формирования  $A \times B$
- Подробности см. книгу Кормена

# Наибольшая общая подстрока

- Даны две строки (first, second)
- Найти их наибольшую общую подстроку
- Решение в лоб?

## Наибольшая общая подстрока

- Даны две строки (first, second)
- Найти их наибольшую общую подстроку
- Решение в лоб?
  
- Тройной цикл: по номеру символа двух строк и общей подстроки



# Итоги

- Рассмотрели
  - Рекуррентный и рекурсивный (разделяй и властвуй) подходы
  - Оценки эффективности
  - Наилучший / наихудший случай
  - Примеры применения на практике
- Далее
  - Алгоритмы сортировки