



# Алгоритмы и структуры данных

Лекция 1. Введение в алгоритмы.

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

2

Зачем всё это?

# Квалификация программиста = ?

- Разработка
  - Предельный объём + сложность + скорость разработки

# Квалификация программиста = ?

- Разработка
  - Предельный объём + сложность + скорость разработки
- Качество кода
  - Читаемость + надёжность + производительность

# Квалификация программиста = ?

- Разработка
  - Предельный объём + сложность + скорость разработки
- Качество кода
  - Читаемость + надёжность + производительность
- Знания
  - Языки + библиотеки + приёмы разработки

# Квалификация программиста = ?

- Разработка
  - Предельный объём + сложность + скорость разработки
- Качество кода
  - Читаемость + надёжность + производительность
- Знания
  - Языки + библиотеки + приёмы разработки
- Анализ
  - Умение разобраться в сложном коде или задаче

# Зачем всё это?

- Разработка
  - Предельный объём + **сложность** + **скорость** разработки
- Качество кода
  - Читаемость + надёжность + **производительность**
- Знания
  - Языки + библиотеки + приёмы разработки
- Анализ
  - **Умение разобратся в сложном коде или задаче**

## Цель курса

- Познакомиться с основами анализа и оценки эффективности алгоритмов (структур)



## Цель курса

- Познакомиться с основами анализа и оценки эффективности алгоритмов (структур)
- Познакомиться с основными группами известных алгоритмов (структур)

## Цель курса

- Познакомиться с основами анализа и оценки эффективности алгоритмов (структур)
- Познакомиться с основными группами известных алгоритмов (структур)
- Научиться применять известные алгоритмы (структуры) при программировании на известном языке программирования

## Цель курса

- Познакомиться с основами анализа и оценки эффективности алгоритмов (структур)
- Познакомиться с основными группами известных алгоритмов (структур)
- Научиться применять известные алгоритмы (структуры) при программировании на известном языке программирования
- Научиться конструировать новые алгоритмы, применяя уже известные идеи

# Литература

- **Томас Кормен и др. Алгоритмы. Построение и анализ. 3-е издание**
- **Никлаус Вирт. Алгоритмы + Структуры данных = Программы**
- **Е. В. Пышкин. Структуры данных и алгоритмы: реализация на C++ (ссылка на странице курса)**
- **McDowell, G. L. Cracking the Coding Interview: 150 Programming Questions and Solutions**
- **S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani. Algorithms**

# Must have!

## © Cracking the Coding Interview

- Data structures
  - Linked Lists
  - Binary Trees
  - Tries (Prefix Trees)
  - Stacks / Queues
  - Array Lists
  - Hash Tables

# Must have!

## © Cracking the Coding Interview

- Algorithms
  - Breadth-First Search
  - Depth-First Search
  - Binary Search
  - Merge Sort
  - Quick Sort
  - Tree Insert / Find / ...

# Must have!

## © Cracking the Coding Interview

- Concepts
  - Bit manipulation
  - Singleton design pattern
  - Factory design pattern
  - *Memory (Stack / Heap)*
  - *Recursion*
  - **Big O**

## Краткое содержание курса (основы)

- Алгоритмы: Big O, рекуррентность, декомпозиция
- Алгоритмы сортировки
- Как создавать алгоритмы?
- Структуры данных: бинарные и префиксные деревья
- Структуры данных: хэш-таблицы
- Графовые алгоритмы
- Динамическое программирование



## Краткое содержание курса (дополнения)

- **Эвристические алгоритмы**
- **Вероятностные алгоритмы**
- **Алгоритмы шифрации**
- **Алгоритмы сжатия данных**
- **Управление памятью**
- **Data Mining**
- **NP-полнота**

# Результат семестра: оценка за проекты + зачёт

- Два из Трёх
  - Индивидуальный проект
  - Упражнения
  - Соревнование
- Средняя оценка из двух лучших:
  - 4.5...5 – «отлично» + «зачёт»
  - 4...4.49 – «хорошо» + «зачёт»
  - 3.5...3.99 – «хорошо» + собеседование по теории
  - 3...3.49 – «удовлетворительно» + собеседование по теории

# Индивидуальный проект

- ▶ Выбирается из списка на странице курса
  - ▶ <http://kspt.icc.spbstu.ru/course/algorithms>
- ▶ Можно выбрать свой, по согласованию с преподавателем
- ▶ Требование: необходимость **сложных** алгоритмов или структур данных при реализации проекта

# Упражнения

- Состоят из набора мелких задач наподобие KotlinAsFirst
- Проверяются через Котоед
- Максимум две итерации проверки + дедлайны
- Расскажу подробнее 18 сентября

# Соревнование

- Модифицированная задача с конкурса ICFPC 2019
- Команды 2-3 человека (одному тоже можно, но это трудно)
- Формулировка задачи и условий: 11-18 сентября
- 2-3 раунда «официальных соревнований» и подведение итогов

# Многообразие задач

- ▶ Интернет
  - ▶ Оптимальные маршруты перемещения данных
  - ▶ Поиск страниц с нужной информацией

# Многообразие задач

- ▶ Интернет
  - ▶ Оптимальные маршруты перемещения данных
  - ▶ Поиск страниц с нужной информацией
- ▶ Электронная коммерция
  - ▶ Шифрование с открытым ключом
  - ▶ Цифровая подпись

# Многообразие задач

- Интернет
  - Оптимальные маршруты перемещения данных
  - Поиск страниц с нужной информацией
- Электронная коммерция
  - Шифрование с открытым ключом
  - Цифровая подпись
- Производство / коммерция
  - Оптимизация выгоды в условиях различных ограничений



# Структуры и алгоритмы

- Структура = данные с определённой организацией
  - Иногда включающие наборы «Инвариантов»
  - Пример: сбалансированное бинарное дерево

# Структуры и алгоритмы

- Структура = данные с определённой организацией
  - Иногда включающие наборы «Инвариантов»
  - Пример: сбалансированное бинарное дерево
- Алгоритм = формальная операция над данными
  - (с определённой организацией)
  - Предусловия / постусловия
  - Сохранение инвариантов

## Пример алгоритма: сортировка

- Input = List<Comparable> in
- Output = List<Comparable> out
- Invariant = None
- Precondition = None
- Postcondition = for any i: out[i+1] >= out[i]

# Корректность алгоритма

- $\text{Output} = \text{Alg}(\text{Input})$
- Математически
  - ЕСЛИ  $\text{Precondition}(\text{Input})$ , ТО  $\text{Postcondition}(\text{Alg}(\text{Input}))$
  - ЕСЛИ  $\text{Invariant}(\text{Input})$ , ТО  $\text{Invariant}(\text{Alg}(\text{Input}))$
- Как доказать корректность?

# Итоги

- Рассмотрели структуру курса и основные понятия
- Далее: основные алгоритмы сортировки
  - Трудоёмкость
  - Ресурсоёмкость
  - Примеры доказательства корректности