



Алгоритмы и структуры данных

Лекция 1. Введение в алгоритмы.

(с) Глухих Михаил Игоревич, glukhikh@mail.ru

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

Квалификация программиста = ?

- Разработка
 - Предельный объём + сложность программы

Квалификация программиста = ?

- Разработка
 - Предельный объём + сложность программы
- Качество кода
 - Читаемость
 - Надёжность
 - Производительность

Квалификация программиста = ?

- Разработка
 - Предельный объём + сложность программы
- Качество кода
 - Читаемость
 - Надёжность
 - Производительность
- Знания
 - Языки + Библиотеки
 - Приёмы разработки

Квалификация программиста = ?

- Разработка
 - Предельный объём + сложность программы
- Качество кода
 - Читаемость
 - Надёжность
 - Производительность
- Знания
 - Языки + Библиотеки
 - Приёмы разработки
- Анализ
 - Умение разобраться в сложном коде / задаче

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
 - 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**

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

Многообразие алгоритмов

- Поиск кратчайшего пути и задача коммивояжёра
- Поиск длиннейшей общей подпоследовательности
- Поиск зависимостей между модулями проекта
- Преобразование Фурье
- Умножение матриц
- Случайная перестановка элементов
- Составление расписаний
- Игра в шахматы
- Распараллеливание

Результат семестра: оценка за проекты + зачёт

- Два из Трёх
 - Индивидуальный проект
 - Упражнения (решение набора относительно простых задач)
 - 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 мной
- Максимум две итерации проверки + дедлайны
- Расскажу подробнее 24 сентября

Соревнование

- Модифицированная задача с ICFPC 2017
- Команды 1-3 человека
- Условие
 - Граф: вершины + важные вершины + дуги
 - Игроки по очереди объявляют дуги своими – пока дуги не закончатся
 - В конце игры начисляются очки: квадрат длины кратчайшего пути для каждой соединённой пары (важная вершина, любая вершина)
- Сервер: <http://kotoed.icc.spbstu.ru:50000/>
- 2-3 раунда «официальных соревнований» и подведение итогов

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