Программа “Алгоритмы” 1 семестр. 2018-2019.

[Введение.](#_8qphmuzwc69)

[Тема 1. Базовые структуры данных](#_n931bnvwpmx0)

[Тема 2. Сортировки и порядковые статистики.](#_cgs0bmjhgqqt)

[Тема 3. Деревья поиска.](#_9zzhsogtje2c)

[Тема 4. Хеш-таблицы.](#_489ximmo2z9s)

[Тема 5. Жадные алгоритмы, Динамическое программирование, Персистентные структуры, Рекуррентные выражения и мастер-теорема.](#_9pkxb8koupci)

# Введение.

1 лекция.

* Определение алгоритма. Примеры простых алгоритмов: вычисление числа Фибоначчи, проверка числа на простоту, быстрое возведение в степень.
* Асимптотические обозначения, работа с ними.
* Определение структуры данных, абстрактного типа данных (интерфейса).
* Массив. Линейный поиск. Бинарный поиск.

# Тема 1. Базовые структуры данных

1 лекция.

* Динамический массив.
* Амортизационный анализ. Амортизированное (учетное) время добавления элемента в динамический массив.
* Двусвязный и односвязный список. Операции. Объединение списков.
* Стек.
* Очередь.
* Дек.
* Хранение стека, очереди и дека в массиве. Циклическая очередь в массиве.
* Хранение стека, очереди и дека в списке.
* Поддержка минимума в стеке.
* Представление очереди в виде двух стеков. Время извлечения элемента.
* Поддержка минимума в очереди.
* Двоичная куча. АТД “Очередь с приоритетом”.

# Тема 2. Сортировки и порядковые статистики.

3 лекции.

* Формулировка задачи. Устойчивость, локальность.
* Квадратичные сортировки: сортировка вставками, выбором.
* Сортировка слиянием.
* Сортировка с помощью кучи.
* Слияние K отсортированных массивов с помощью кучи.
* Нижняя оценка времени работы для сортировок сравнением.
* Быстрая сортировка. Выбор опорного элемента. Доказательство среднего времени работы.
* TimSort.
* Сортировка подсчетом. Карманная сортировка.
* Поразрядная сортировка.
* MSD, LSD. Сортировка строк.
* Binary QuickSort.
* Поиск k-ой порядковой статистики методом QuickSelect (без доказательства среднего времени работы).
* Поиск k-ой порядковой статистики за линейное время.

# Тема 3. Деревья поиска.

4 лекции.

* Определение дерева, дерева с корнем. Высота дерева, родительские, дочерние узлы, листья. Количество ребер.
* Обходы в глубину. pre-order, post-order и in-order для бинарных деревьев.
* Обход в ширину.
* Дерево поиска.
* Поиск ключа, вставка, удаление.
* Необходимость балансировки. Три типа самобалансирующихся деревьев.
* Декартово дерево. Оценка средней высоты декартового дерева при случайных приоритетах (без доказательства).
* Построение за O(n), если ключи упорядочены.
* Основные операции над декартовым деревом.
* АВЛ-дерево. Вращения.
* Оценка высоты АВЛ-дерева.
* Операции вставки и удаления в АВЛ-дереве.
* Красно-черное дерево.
* Оценка высоты красно-черного дерева.
* Операции вставки и удаления в красно-черном дереве.
* Сплей-дерево. Операция Splay.
* Поиск, вставка, удаление в сплей-дереве.
* Учетная оценка операций в сплей-дереве = O(log n) без доказательства.
* B-деревья.

# Тема 4. Хеш-таблицы.

2 лекции.

* Хеш-функции. Остаток от деления, мультипликативная.
* Деление многочленов - CRC.
* Обзор криптографических хеш-функций. CRC\*, MD\*, SHA\*.
* Полиномиальная. Ее использование для строк. Метод Горнера для уменьшения количества операций умножения при ее вычислении.
* Хеш-таблицы. Понятие коллизии.
* Метод цепочек (открытое хеширование).
* Метод прямой адресации (закрытое хеширование).
* Линейное пробирование. Проблема кластеризации.
* Квадратичное пробирование.
* Двойное хеширование.

# Тема 5. Жадные алгоритмы, Динамическое программирование, Персистентные структуры, Рекуррентные выражения и мастер-теорема.

4 лекции.

* Общая идея жадных алгоритмов.
* Задача о рюкзаке.
* Общая идея последовательного вычисления зависимых величин. Идея введения подзадач (декомпозиции) для решения поставленной задачи. Восходящее ДП. Нисходящее ДП, кэширование результатов.
* Вычисление чисел Фибоначчи.
* Нахождение наибольшей возрастающей подпоследовательности за O(N2) и за O(N log N).
* Количество способов разложить число N на слагаемые.
* Количество способов разложить число N на различные слагаемые.
* Нахождение наибольшей общей подпоследовательности.
* Методы восстановления ответа в задачах динамического программирования.
* Расстояние Левенштейна.
* Персистентные структуры данных. Версии, возможность модифицировать любую версию. Примеры использования.
* Персистентный стек.
* Персистентное дерево поиска. Добавление, удаление узла. Повороты.
* Рекуррентные выражения. Способы доказательства оценок: метод подстановки и метод разворачивания суммы.
* Мастер-теорема (без доказательства).