

Курс «Алгоритмы и алгоритмические языки»

Лекция 24

Обзор курса.

Часть I - Формальные системы описания алгоритмов

- Задачи обработки информации и алгоритмы.
- Неформальное (интуитивное) определение алгоритма.
 - Конечность
 - Определенность
 - Простота и понятность
 - Массовость
- Формализация алгоритма.
 - Алфавит, слова
 - Кодирование
- Машина Тьюринга.
 - Определение
 - Вычислимость по Тьюрингу

Обзор курса.

Часть I - Формальные системы описания алгоритмов

- Способы представления машин Тьюринга. Нормальные вычисления.
 - Эквивалентность нормальных МТ и общего определения МТ
- Диаграммы Тьюринга. Построение диаграмм Тьюринга.
 - Элементарные действия и МТ в диаграммах
 - Композиция диаграмм
 - Построение таблиц по диаграммам
- Понятие универсальной машины Тьюринга. Построение универсальной машины Тьюринга.
 - Моделирование МТ

Обзор курса.

Часть I - Формальные системы описания алгоритмов

- Проблема останова и алгоритмическая неразрешимость.
- Алгоритмическая неразрешимость проблемы самоприменимости.
 - Доказательства от противного
- Тезис Тьюринга – Черча.
- Нормальные алгоритмы Маркова.
 - Определение и примеры
 - Тезис Маркова
- Эквивалентность формальных систем описания алгоритмов.
 - Критика МТ

Обзор курса.

Часть II - Язык программирования Си

- Язык программирования Си. Описание Си-машины.
 - Куча, стек, статические данные
 - Процессор и внешние устройства
- Структура Си-программы.
 - Переменные и функции
- Типы данных.
 - Базовые типы, размеры
- Переменные, константы. Классы памяти. Области видимости и существования переменных.
 - Локальные, глобальные переменные
 - Инициализация переменных
 - Способы задания констант

Обзор курса.

Часть II - Язык программирования Си

- Арифметические типы данных. Арифметические операции над целочисленными данными.
 - Операция присваивания, побочные эффекты
 - Точки следования
- Приведение типов. Правила неявного преобразования типов («обычные арифметические преобразования»).
- Старшинство операций.
- Операторы языка Си: условные операторы, операторы цикла, операторы перехода, составной оператор (блок).

Обзор курса.

Часть II - Язык программирования Си

- Символьный тип данных (char).
 - Символьные константы
 - Представление букв и цифр в ASCII
- Массивы и строки.
 - Инициализация массивов
 - Использование '\0' как признака конца строки
- Указатели. Адресная арифметика. Преобразование типа указателя.
 - Операции разыменования и взятия адреса
 - Сложение с числом, вычитание числа, вычитание указателей
 - Отношение и сравнение
 - void *
 - Преобразование типа
 - Многомерные указатели

Обзор курса.

Часть II - Язык программирования Си

- Указатели и массивы. Массивы указателей. Многомерные массивы.
 - Имя массива – указатель на начало
 - Переменную массива нельзя менять, указатель – можно
 - Индексирование указателей
 - Указатель на массив и массивы указателей
 - Многомерные массивы (точнее, их отсутствие)
- Функции. Объявление функции. Формальные параметры. Возвращаемое значение. Побочный эффект функции. Функции типа void. Определение функции.
 - Оператор return

Обзор курса.

Часть II - Язык программирования Си

- Библиотечные функции. Вызов функции. Фактические параметры и способ их передачи (по значению).
 - Передача параметров по значению
 - Указатели для модификации значения функцией
- Указатели и параметры функции. Передача массива в функцию. Квалификатор **restrict**.
- Рекурсивные функции. Квалификатор **inline**.
 - Рекурсия и итерация
 - Подстановка тел функций в точку вызова (встраивание)

Обзор курса.

Часть II - Язык программирования Си

- Операция **sizeof**.
 - Применимость к полностью определенным типам
- Арифметические операции над данными с плавающей точкой.
 - Дробные двоичные числа
 - Мантисса, порядок
 - Нормализованные и денормализованные числа
 - Как складывать и вычитать
- Отношения и логические операции. Поразрядные операции. Реализация абстрактного типа данных «множество».
 - Беззнаковые типы и поразрядные операции

Обзор курса.

Часть II - Язык программирования Си

- Структуры, перечисления, объединения. Указатели на структуры. Битовые поля. Ключевое слово **typedef**.
 - Объявление и инициализация структур
 - Размеры структур и объединений
 - Операция ->
 - Использование перечислений вместо макро-констант
- Динамическое распределение памяти.
 - malloc, calloc, realloc, free, alloca
- Указатель на функцию.
 - Вызов по указателю
 - Передача указателей на функции как аргументов

Обзор курса.

Часть II - Язык программирования Си

- Сборка Си-программы: препроцессирование, компиляция, компоновка. Директивы препроцессора. Директива `#include` и заголовочные файлы. Условная компиляция.
 - `.h` для интерфейса и `.c` для реализации (`static`, `extern`)
 - Этапы компиляции
- Стандартные библиотеки языка Си.
 - Ввод-вывод
 - Работа с символьными типами и строками
 - Работа с целыми числами и числами с плавающей точкой
 - Динамическое выделение памяти, сортировка

Обзор курса.

Часть III - Алгоритмы и структуры данных

- **Динамические структуры данных. Список (однонаправленный и двунаправленный).**
 - Список нельзя индексировать как массив
 - Нужно рассматривать случаи модификации указателя на голову списка (необходимость двойного указателя)
 - Добавление элемента в конец и в начало списка
- **Стек и его реализация на массиве и на списке. Очередь.**
 - Стек/очередь как абстрактные структуры данных (доступ через интерфейсные функции без деталей реализации)
- **Применение стека для преобразования выражений в польскую запись.**
- **Топологическая сортировка узлов ациклического ориентированного графа: постановка задачи, алгоритм Вирта.**

Обзор курса.

Часть III - Алгоритмы и структуры данных

- Сортировка. Основные алгоритмы сортировки. Оценка сложности алгоритмов сортировки.
 - Внешняя и внутренняя сортировки
 - Сортировка выбором, вставками, пузырьком
 - Сортировка сравнением невозможна быстрее, чем за $O(n \log n)$
- Быстрая сортировка Хоара.
 - Библиотечная функция `qsort`
 - Стратегия “разделяй и властвуй”
 - Сложность $O(n \log n)$, наихудший случай $O(n^2)$
- Двоичное дерево. Представление двоичного дерева в памяти компьютера. Способы обхода двоичного дерева и их рекурсивная и нерекурсивная реализации.
 - Прямой обход – топологический порядок
 - Необходимость использования итерации вместо рекурсии

Обзор курса.

Часть III - Алгоритмы и структуры данных

- Прошитое двоичное дерево. Прошитое двоичное дерево с заголовком.
 - Сохранение следующего/предыдущего элемента вместо пустых ссылок на детей
 - Экономия времени при многократных обходах
 - Идея служебного заголовка для упрощения кода
- Двоичные деревья поиска. Реализация словарных операций: *search* (найти), *insert* (вставить) и *delete* (удалить). Реализация операций *min* (минимум), *max* (максимум), *pred* (предыдущий) и *succ* (следующий). Построение двоичного дерева поиска.
 - Определение связывает значения элементов в корне и во всем поддереве
 - Линейная сложность базовых операций от высоты
 - Важность балансировки (минимальная высота)

Обзор курса.

Часть III - Алгоритмы и структуры данных

- Структура данных «пирамида». Пирамидальная сортировка.
 - Использование идей деревьев поиска в сортировке
 - Не требуется дополнительная память для дерева
 - Наихудшая сложность $O(n \log n)$
- Сбалансированные двоичные деревья. Деревья Фибоначчи. Число узлов в дереве Фибоначчи высоты h .
- AVL-деревья. Базовые операции над AVL-деревьями (удалить дерево, поиск по ключу, минимальный ключ, максимальный ключ) и их реализация.
 - Сохраняются базовые операции как над деревьями поиска
 - AVL-деревья по “качеству” находятся между деревьями Фибоначчи и полностью сбалансированными

Обзор курса.

Часть III - Алгоритмы и структуры данных

- **Балансирование AVL-деревьев. Реализация операции вставки узла в AVL-дерево. Построение AVL-дерева.**
 - Сведение к базовым поворотам
 - Реализация заключается в основном в перевешивании ссылок
- **Красно-черные деревья. Вставка узла в красно-черное дерево.**
 - Чередование двух цветов позволяет иметь высоту не более $2 \log_2 n$
 - Схожесть подхода балансировки с AVL-деревьями: иногда можно обойтись перекраской, но нужны и повороты

Обзор курса.

Часть III - Алгоритмы и структуры данных

- Словарные операции и их реализация с помощью хеш-функций. Методы построения хеш-функций.
 - Борьба за равномерность хеширования и минимизацию коллизий
 - Нужно хорошее “перемешивание” бит ключа
- Хеширование с цепочками. Хеширование с открытой адресацией. Двойное хеширование.
 - Открытая адресация позволяет избежать расходов на дополнительный уровень косвенности (список)
 - Двойное хеширование наиболее равномерно разбрасывает элементы и избегает кластеров
 - Усложнение удаления
 - Использование чисел на местах хранения указателей
- Оценка среднего времени успешного поиска в хеш-таблице с коэффициентом заполнения α .

Обзор курса.

Часть III - Алгоритмы и структуры данных

- Цифровой поиск.
 - Ключи рассматриваются как последовательность символов
 - Ветви по m значениям позволяют получить сложность $O(\log_m N)$
- Поиск подстрок по образцу. Алгоритм Кнута – Морриса – Пратта.
 - Терминология: текст, образец, префикс, суффикс.
 - Префикс-функция – насколько можно сдвинуться исходя из имеющихся данных о тексте без потери вхождений образца
 - Сложность $O(n+m)$
- Алгоритмы перебора множеств. Рекурсивный алгоритм генерации перестановок. Алгоритм Нарайаны.
 - Применимость алгоритмов перебора при программировании
 - Алгоритм Нарайаны как итеративная версия рекурсивного алгоритма (лексикографический порядок)