

Теория:

1. $O(n)$ нотация

Что такое $O(n)$, $\Omega(n)$ и $\Theta(n)$ нотации, математическое определение и смысл. Знать(или уметь выводить) сложности всех пройденных алгоритмов: алгоритм бинарного поиска в отсортированном массиве, алгоритмы сортировки (пузырьком, вставками, выбором, быстрая, сортировка слиянием, подсчётом, цифровая). Знать или уметь выводить сложности операций с пройденными структурами данных: поиск массива по индексу в массиве и списке, вставка в массив(статический и динамический) и список; удаление из массива и списка; поиск по массиву и списку.

2. Структуры данных. Статический и динамический массив. Связный список

Что такое структура данных. Статический массив. Объявление, инициализация и использование статического массива в языках C и C++. Динамический массив. Объявление, инициализация и использование динамического массива в языках C и C++. Операции доступа к элементу, добавления(в начало, середину и конец) и удаления из массива и их сложности. Связный список. Как реализуются функции нахождения длины списка, вставки элемента в начало/конец, удаления элемента из начала/конца, поиска в списке, обращения списка. Алгоритмические сложности операций со списком. Двусвязный список. Использование динамического массива в языке C++, стандартный контейнер `vector`. Использование связного списка в языке C++, стандартный контейнер `list`.

3. Структуры данных. Графы и деревья.

Что такое структура данных. Определения графа. Связный граф, взвешенный граф, простой граф, ациклический граф, дерево. Представления графа с помощью списка смежных вершин и матрицы смежности. Обход в глубину и ширину. Алгоритм Дейкстры. Использование очереди с приоритетом в алгоритме Дейкстры. Деревья и их реализация на языке C. Бинарные деревья поиска и их реализация на языке C. Проход по дереву. Операции поиска, добавления и удаления элемента в дерево. Их сложность и реализация на языке C.

4. Структуры данных. Хэш-таблица.

Что такое хэш функция. Свойства хорошей хэш-функции. Что такое хэш-таблица. Коллизии, метод цепочек и открытая адресация. Операцию добавления элемента в хэш-таблицу, операцию поиска и операцию удаления элемента в хэш-таблице и их сложность. *Примеры различных хэш-функций. *Хэш-функции основанные на делении. *Хэш-функция Пирсона.

5. Абстрактные типы данных. Стек, очередь и очередь с приоритетом.

Что такое абстрактный тип данных. Различие между структурой данных и абстрактным типом данных. Стек. Операции `push` и `pop`. Реализация стека на основе массива. Динамическое выделение памяти для стека. Стратегии перевыделения памяти(аддитивная и мультипликативная). Очередь. Реализация стека и очереди на основе связного списка. Алгоритмические сложности операций со стеком и очередью. Очередь с приоритетом. *Двоичная куча. *Реализация очереди с приоритетом с помощью двоичной кучи.

6. Абстрактные типы данных. Множество и словарь.

Множество. Реализация множества с помощью дерева и с помощью хэш-таблицы. Операции поиска элемента в множестве, добавления и удаления элемента в множестве. Использование множеств в языке C++(контейнеры библиотеки STL: `set`, `multiset`, `unordered_set` и `unordered_multiset`). Словарь (или ассоциативный массив). Пара Ключ-Значение. Реализация словаря с помощью дерева и с помощью хэш-таблицы. Операции поиска элемента в множестве, добавления и удаления элемента в словаре. Использование словарей в языке C++(контейнеры библиотеки STL: `map`, `multimap`, `unordered_map` и `unordered_multimap`).

7. Сортировка

Знать сортировки пузырьком, вставками, выбором, быструю сортировку, сортировку слиянием, подсчётом и сортировка выбором. Алгоритмические сложности всех этих алгоритмов. Уметь писать сортировку выбором, быструю сортировку и сортировку подсчётом. Парадигма разделяй и властвуй. Логарифмическая сложность. Алгоритм бинарного поиска в отсортированном массиве, сортировка слиянием, быстрая сортировка.

8. Память

Что такое переменная. Адреса переменных. Указатели. Определение адреса переменной. Разыменование. Основные типы данных и их размеры: `int`, `float`, `double`, `char`, `void*`, `int*`, `char*`. Адресная арифметика. Динамическое выделение и освобождение памяти в куче: `malloc()`, `free()` и `realloc()`. Утечки памяти. Указатель `void*`.

9. **Передача аргументов в функцию в С и С++**
Все типы передачи аргументов в функцию. Передача по ссылке и передача по значению. Передача через адрес переменной. Передача через указатель на константу. Ссылки С++. Передача по ссылке(reference С++). Передача через константную ссылку(С++). Передача массивов, строк и двумерных массивов в функцию. Передача структур в функцию. Возвращение аргумента. Возвращение нескольких аргументов из функции. Возвращение массива из функции. Ключевое слово void.
10. **Сегменты памяти**
Что такое сегменты памяти. Сегмент памяти стек вызовов(или просто стек). Выделение памяти в стеке. Переполнение стека. Сегмент памяти куча(heap). Динамическое выделение и освобождение памяти в куче: malloc(), free() и realloc(). Динамическое выделение двумерного массива. Преимущества и недостатки кучи перед стеком. Сегмент памяти text. Ошибка Segmentation Fault.
11. **Этапы компиляции. Сборка проекта.**
Что такое файл исходного кода и исполняемый файл. Этап компиляции: препроцессинг, компиляция и линковка. Директивы препроцессора #include и #define. Компиляция программы с помощью gcc. Опции gcc: -E, -c. Header-файлы. Раздельная компиляция, её преимущество. Make-файлы. Сmake.
12. **Побитовые операции. Представление базовых типов в памяти компьютера.**
Побитовые операции: побитовые И, ИЛИ, НЕ, исключающее ИЛИ. Побитовые сдвиги влево и вправо. Умножение числа на степень двойки с помощью побитовых сдвигов. Представление беззнакового целого числа(unsigned int) в памяти компьютера. Представление знакового целого числа(int) в памяти компьютера, дополнительный код. Представление вещественного числа(float) в памяти компьютера. IEEE 754. Порядок байтов, Little и Big Endian.
13. **С++ features**
Ссылки(reference), передача аргументов в функцию с помощью ссылок и с помощью константных ссылок. Ввод и вывод в С++, cin и cout. Перегрузка функций в С++. Значения аргументов функций по умолчанию в С++. Операторы new/delete. Их преимущества по сравнению с malloc/free. Класс string. Строки С++(string) и строки в стиле С (char*), их различия.
Дополнительные фишки С++11. Ключевое слово auto. Обобщённая инициализация(с помощью {}). Range-based for loop.
14. **Объектно-ориентированное программирование (ООП). Основы.**
Основные принципы ООП: инкапсуляция, наследование и полиморфизм. Элементы ООП в языке С. Классы С++. Поля и методы класса. Модификаторы доступа public и private. Различие между ключевыми словами struct и class. Преимущества закрытых(private) полей и методов. Конструкторы. Перегрузка конструкторов. Конструктор по умолчанию. Список инициализации конструктора. Деструкторы. RAII. Скрытый указатель this. Ключевое слово static. Статические поля класса. Ключевое слово friend. Дружественные функции и классы. Перегрузка операторов в С++. Преимущество перегрузки операторов. Перегрузка operator+, operator*, operator>, operator«.
15. **Объектно-ориентированное программирование (ООП). Наследование.**
Зачем нужно наследование? Наследование в С++. Модификатор доступа protected. Добавление новых полей и методов в классе наследнике. Перегрузка функций в классе наследнике. Работа с указателем базового класса на экземпляр класса-наследника. Виртуальные функции и ключевое слово virtual. Таблица виртуальных методов. Абстрактные виртуальные функции. Абстрактные виртуальные классы. Интерфейсы. Виртуальный деструктор.
16. **Шаблоны.**
Шаблонные функции. Шаблоны классов. Использование шаблонов функций и классов. Переменная, как шаблонный параметр. Библиотека STL. Шаблонный класс pair. Шаблонные контейнеры vector, set, list, map. Итераторы STL. Проход по контейнеру с помощью итераторов. Стандартные функции find() и sort().

Материалы для подготовки:

1-й семестр:

1. Кормен, Лейзерсон, Ривест. Алгоритмы: построение и анализ.
lib.mipt.ru/book/16736/
2. Керниган Ритчи Язык программирования C
lib.mipt.ru/book/266005/
3. Простое введение в алгоритмы hexlet.io.
youtube.com/watch?v=8JITwMg1dyw&list=PLwwk4BNih4fjIT5cT4i1s93b99aJScUGB
4. Лекторий МФТИ:
lectoriy.mipt.ru/course/ComputerTechnology-Informatics-14L#lectures
5. Курс по структурам данных stepic.org
youtube.com/watch?v=vRvSdWVst54
6. Продвинутый уровень. Язык C и структуры данных в Йелле.
www.cs.yale.edu/homes/aspnes/classes/223/notes.html

2-й семестр:

1. Кормен, Лейзерсон, Ривест. Алгоритмы: построение и анализ.
lib.mipt.ru/book/16736/
2. [learncpp](http://learncpp.com/):
<http://www.learncpp.com/>
3. Код с лекций:
<https://github.com/avasyukov/oop-2nd-term/tree/master/2018>

Дополнительные материалы по некоторым вопросам:

- Вопрос №10. Сегменты памяти:
youtube.com/watch?v=_8-ht2AKyH4 (англ.)
tproger.ru/translations/programming-concepts-stack-and-heap/
- Вопрос №11. Этапы компиляции:
youtube.com/watch?v=y1A55D4B4_M