

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. Стек и очередь

Структуры данных и абстрактный типы 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.

3. Связный список

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.

4. 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. Уметь использовать стандартный функцию сортировки `qsort`.

5. Разделяй и властвуй

Парадигма разделяй и властвуй. Логарифмическая сложность. Алгоритм бинарного поиска в отсортированном массиве, сортировка слиянием, быстрая сортировка. Уметь писать бинарный поиск и быструю сортировку.

6. 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*`. Преобразование типа указателя. Организация массива в памяти. Организация связного списка в памяти.

7. Передача аргументов в функцию

Все типы передачи аргументов в функцию. Передача по ссылке и передача по значению. Передача через адрес переменной. Передача через указатель на константу. 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`.

8. 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`.

9. Этапы компиляции

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`.

Практика:

Из того, что не было на контрольной работе:

1. **Основные команды командной строки linux:** `cd`, `ls`, `pwd`, `cp`, `mv`, `mkdir`, текстовый редактор `nano` (или `vi`), компилятор `gcc`. Создать файл исходного кода программы, которая будет считывать числа из входного файла, складывать их и записывать результат в другой файл. Создать файл входных данных. Скомпилировать программу. Всё это, пользуясь только командами командной строки. Для тех, у кого нет возможности использовать linux:
www.tutorialspoint.com/unix_terminal_online.php
Там нет редактора `nano`, но есть редактор `vi` (или `vim`).
Основы работы с `vi` вы можете посмотреть тут: youtube.com/watch?v=R33F0EDivwk
2. **Аргументы командной строки, бинарное чтение/запись**
Уметь использовать в программе аргументы командной строки: `argc`, `argv`. Бинарное чтение/запись – `fread`, `fwrite`.

Материалы для подготовки:

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=PLwwk4BHih4fjIT5cT4i1s93b99aJScUGB
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/aspnec/classes/223/notes.html

Дополнительные материалы по некоторым вопросам:

- Вопрос №8. Сегменты памяти:
youtube.com/watch?v=_8-ht2AKyH4 (англ.)
tproger.ru/translations/programming-concepts-stack-and-heap/
- Вопрос №9. Этапы компиляции:
youtube.com/watch?v=y1A55D4B4_M