

# Список вопросов к теоретическому зачёту ЛКШ.С'

## 1. Язык Python

- a. Целые числа и числа с плавающей точкой
- b. Арифметические операции
- c. Оператор if
- d. Операции сравнения и логические операции
- e. Булевский тип True/False
- f. Циклы for, while. Операторы break, else
- g. Строки, операции со строками (join, split, \*, +)
- h. Списки, операции со списками
- i. Генераторы списков
- j. Параллельное присваивание
- k. Ввод/вывод с клавиатуры и из файлов
- l. Синтаксис функций
- m. Области видимости переменных в функциях
- n. Значения по умолчанию, обращение к параметру по имени
- o. Стек вызовов функций. Ограничение на глубину стека в Python
- p. Метод sort, его параметр key
- q. Двумерные массивы. Считывание, создание, вывод

## 2. Асимптотическое время работы программы/функции, O-символика.

## 3. Арифметика

- a. Проверка числа на простоту
- b. Эффективное нахождение самого маленького и самого большого собственного делителя
- c. Разложение числа на простые множители
- d. Наибольший общий делитель, наименьшее общее кратное.
- e. Алгоритм Евклида
- f. Решето Эратосфена

## 4. Линейный поиск. Поиск двух минимумов одним проходом

## 5. Сортировки

- a. Квадратичные сортировки
  - i. Пузырьком
  - ii. Вставками
  - iii. Выбором
- b. Сортировка подсчётом
- c. Устойчивость сортировок

## 6. Рекурсия

- a. База, рекуррентное соотношение.
- b. Время работы рекурсивных функций
- c. Ханойские башни
- d. Алгоритм сортировки слиянием, время работы
- e. Рекурсивный перебор всех подмножеств
- f. Рекурсивный перебор подмножеств размера k

## 7. Бинарный поиск

- a. Алгоритм бинарного поиска, время работы
- b. Поиск первого/последнего вхождения элемента
- c. Бинарный поиск по ответу

## 8. Основные понятия теории графов

- a. Граф, степень вершины, входящая и исходящая степень, истоки и стоки, путь, простой путь, цикл, простой цикл, компоненты связности, висячие вершины
- b. Виды графов: связные/несвязные, циклические/ациклические, ориентированные/неориентированные, полные, взвешенные, деревья, k-ичные деревья
- c. Хранение графов: матрица смежности, список смежности, список рёбер.
- d. Связь между количеством вершин и рёбер в дереве, лесе.
- e. Существование висячей вершины в дереве.
- f. Теорема о рукопожатиях (количество вершин нечётной степени чётно)

## 9. Поиск в глубину

- a. Обход в глубину
- b. Оценка сложности с разными структурами для хранения графа
- c. Применения поиска в глубину:
  - i. Подсчёт количества компонент связности
  - ii. Подсчёт количества вершин в каждой компоненте связности
  - iii. Проверка наличия цикла (ориентированный и неориентированный случай)
- d. Восстановление пути

## 10. Поиск в ширину

- a. Обход в ширину
- b. Оценка сложности с разными структурами для хранения графа
- c. Нахождение кратчайшего пути на невзвешенном графе
- d. Восстановление пути

## 11. Динамическое программирование

- a. Правила динамического программирования: подзадача, рекуррентное соотношение, начальное значение, порядок вычисления, получение ответа
- b. Мячик на лестнице
- c. Платная лестница
- d. Черепашка с препятствиями
- e. Черепашка с монетами. Контрпример к жадному алгоритму. Восстановление ответа

## 12. Динамика «длина + последний»

- a. Количество последовательностей без двух единиц подряд
- b. Телефонные номера, получаемые ходом коня

## 13. Задача о рюкзаке

- a. Непрерывный рюкзак
- b. Рюкзак со стоимостями

## 14. Контейнеры в Python

- a. Структуры данных очередь, стек, дек. Deque в Python, методы `appendleft`, `popleft`, `append`, `pop`
- b. Структура данных `set`, инициализация (`set()`, генератор), методы `add`, `remove`, `discard`, функция `len`
- c. Структура данных `dict`, инициализация, обращение к элементам, создание нового ключа, методы `items`, `keys`, `values`, `get`
- d. Функция `hash`, хешируемые и нехешируемые объекты, устройство `set` и `dict`