

Правила проведения зачета

по дисциплине «Алгоритмы и структуры данных»

1. Зачет по дисциплине «Алгоритмы и структуры данных» в 2019 году пройдет в письменной форме, 24 декабря (во вторник (во время лекции)), результаты и подведение итогов 25 декабря (в среду (во время лабораторной)).
2. Билет к зачету будет содержать два теоретических вопроса и два практических задания (задачи).
3. К выполнению задания допускаются все студенты без исключения. Зачет получают студенты, успешно выполнившие все виды работ (лабораторные, РГР, зачет).

Список теоретических вопросов и типов задач

к зачету по дисциплине «Алгоритмы и структуры данных»

Вопросы:

1. Виды асимптотических оценок, наиболее распространенные классы сложности.
2. Списки и массивы: преимущества и недостатки. Стек и очереди: способы реализации.
3. Деревья: определение, способы реализации, виды обходов.
4. Графы: определение, способы реализации.
5. Отображения и префиксные деревья: назначение, способы реализации.
6. Двоичные деревья поиска: виды и их основные свойства.
7. Недвоичные деревья поиска: виды и их основные свойства.
8. Хеш-таблица определение, назначение. Открытое (внешнее) хеширование.
9. Требования к хеш-функциям, примеры хеш-функций для различных типов данных.
10. Способы разрешения коллизий при закрытом (внутреннем) хешировании.
11. Устойчивость сортировки, пример устойчивого и неустойчивого алгоритма.
12. Пирамидальная сортировка – алгоритм, реализация пирамиды на массиве и работа с ней.
13. Сортировка слиянием: идея метода, временная сложность, применение.
14. Карманная сортировка: идея метода, временная сложность, применение.
15. Сортировка подсчетом – алгоритм.
16. Префикс-функция строки: определение, применение для поиска подстроки.
17. Вычисление префикс-функции: алгоритм, доказательство линейной оценки.
18. Алгоритм Рабина-Карпа.
19. Алгоритм Кнута-Морриса-Пратта.
20. Алгоритм Ахо-Корасик: постановка задачи, укрупненный алгоритм, время работы.
21. Алгоритм Ахо-Корасик: этапы построения автомата.
22. Суффиксные деревья: определение, свойства, область применения.
23. Наивные алгоритмы построения суффиксных деревьев.
24. Ключевые идеи и условные обозначения в алгоритме Маккрейта построения суффиксного дерева.
25. Ключевые идеи и условные обозначения в алгоритме Укконена построения суффиксного дерева.
26. Суффиксный массив: построение с использованием деревьев, поиск подстроки – алгоритм.

27. Построение суффиксного массива без использования деревьев. Продемонстрировать ключевые идеи алгоритма на примере.
28. Поиск наибольшей общей подпоследовательности двух строк методом динамического программирования – алгоритм.
29. Ключевая идея алгоритма Хиршберга поиска LCS.
30. ~~Ключевые идеи и условные обозначения в алгоритме Ханга-Шиманского поиска LCS.~~
31. Сжатие данных без потерь. Энтропийное кодирование (на примере некоторого конкретного алгоритма).
32. Сжатие данных без потерь. Словарные методы кодирования (на примере некоторого конкретного алгоритма).

Задачи:

1. Определить вычислительную сложность алгоритма по блок-схеме.
2. Записать последовательность посещения вершин при заданном способе обхода заданного дерева.
3. Представить заданный граф заданным способом.
4. Построить префиксное дерево для заданного набора строк.
5. Добавить/удалить значение в заданное АВЛ-дерево.
6. Добавить/удалить значение в заданное 2-3 дерево.
7. Добавить/удалить значение заданную хеш-таблицу при линейном зондировании.
8. Показать последовательность изменений при сортировке «пузырьком» заданного набора чисел.
9. Показать последовательность изменений при сортировке «вставками» заданного набора чисел.
10. Показать последовательность изменений при «поразрядной» сортировке заданного набора чисел.
11. Показать последовательность изменений при «быстрой» сортировке заданного набора чисел.
12. Записать префикс-функцию для заданной строки.
13. Построить автомат Ахо-Корасик для заданного словаря.
14. Построить суффиксное дерево для заданной строки.
15. Поиск LCS заданных строк методом динамического программирования: заполнить массив.
16. ~~Заполнить массив пороговых значений алгоритма Ханга-Шиманского для заданных строк.~~
17. Построить дерево Хаффмана для заданной строки.
18. Восстановить исходное сообщение по заданной таблице частот и коду, полученному арифметическим кодированием.
19. Закодировать заданную строку методом LZ77.
20. Закодировать заданную строку методом LZ78.