Массив не отсортирован

Есть 3 алгоритма

1. Быстрая сортировка O(N\*log(N)) –

Вход – массив, выход – отсортированный массив

1. Бинарный поиск O(log(N))

Вход - отсортированный массив, выход – искомый элемент

1. Линейный поиск O(N)

Вход – массив, выход – искомый элемент

======

Предложите способ поиска одного элемента в массиве с наименьшей вычислительной сложностью.

Массив на 1024 элементов

Возможные предыдущие расширения

512+256+128+…+1~1024

Оформление отчетов:

Делается общий отчет, но его надо сдавать частично, на контрольных неделях. В отчете указывается название дисциплины и преподаватель.

Для каждой главы делается отдельный параграф.

Для теоретических заданий пишется постановка, ответ и обоснование

Для заданий, которые требуют написания кода пишется постановка, код, объяснение и результат выполнения кода. При необходимости описывается анализ результатов кода (если он требуется по заданию).

Глава 1.

Задание на 4 – упражнения из главы 1.

Задание на 5 – дополнительно решить

Имеется массив объектов отсортированный по 3 и более критериям. Осуществить в нем бинарный поиск

Глава 2.

Задание на 4 – упражнения из главы 2.

Задание на 5 – в 2.2. известно, что список не будет длиннее 1000 элементов.

Напишите код для алгоритма, такой чтобы вычислительная сложность на одно действие (чтение, запись) была O(1). Можно использовать любые структуры данных, при условии, что можете обосновать выбор. (Структура должна быть реализована вами)

Глава 3.

Задание на 4 - упражнения из главы 3.

Реализовать метод вычисления члена последовательности Фибоначчи на основе рекурсии. Определить вычислительную сложность данного алгоритма и подтвердить это экспериментально. Проверить это на стандартной реализации длинной арифметики и вашей реализации на основе сложения столбиком.

Задание на 5 – вычисление последовательности Фибоначчи с использованием длинной арифметики. Метод должен работать без погрешности. Размер последовательности такой, что время работы программы не менее 10 секунд.

Сделать выводы о вычислительной сложности алгоритма, имея ввиду то, что время операции сложения зависит от количества разрядов, хранимых в памяти.

Сделать одну реализацию, основанную на встроенных структурах данных и 1 на своих.

Глава 4.

На 4 – упражнения из главы 4

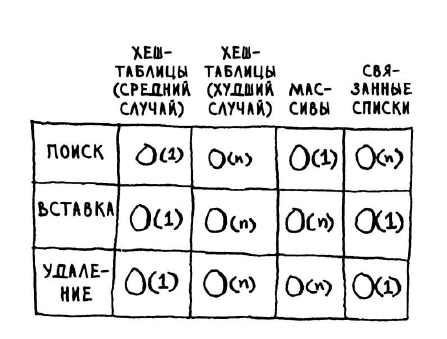
На 5 – Определите вычислительную сложность сортировки подсчётом.

Реализуйте данный алгоритм. Выполните экспериментальное определение вычислительной сложности. Подберите данные, при которых ваша реализация быстрее стандартной быстрой сортировки.

Глава 5.

упражнения из главы 5

Определить время вычислений для каждой структуры данных экспериментально. Сравнить с теоретическими данными.



Конец контрольной недели

Глава 6.

Упражнения из главы 6.

6.1, 6.2 – код и вручную

В примере из учебника добавить дополнительную структуру, в которой устанавливается соответствие между именами людей и фактом того, что они являются продавцами манго. Эта структура должна использоваться для проверки (вместо заглушки – проверки по последней букве).

Задание на 5.

Проверить, можно ли дойти от начальной точки лабиринта до конечной, (движение разрешено по вертикали, горизонтали на 1 клетку).

**Данные на входе.**

Массив логического типа. Истина – клетка свободная, ложь – клетка занята.

Координаты начала пути x1, y1

Координаты конца пути x2,y2

**Данные на выходе.**

-2 – Начальная или конечная точка не соответствуют свободной клетке массива

-1 – Пути от начальной до конечной клетки нет

Н – длина пути от начальной до конечной клетки

Глава 7.

Упражнение 7.1.

Для графа A на примере кода показать, как код из книги (или его эквивалент на предпочитаемом вами языке программирования) находит решение. Приготовиться объяснить любой шаг алгоритма.

Для графов B,C выполнить решение вручную.

Показать проблему с циклами в алгоритме Дейкстры и предложить ее решение.

На оценку 5.

Нужно написать код для задачи

<https://acmp.ru/index.asp?main=task&id_task=139>

Протестировать код на графах из упражнения 7.1, а также на графах, полученных из исходных путем умножения весов всех ребер на -1.

Глава 8.

Задание на 4. Реализовать жадный алгоритм для задачи о рюкзаке из главы 8. В качестве критерия выбора предмета рассмотреть стоимость за единицу места в рюкзаке, а не стоимость предмета, как это сделано в книге.

Задание на 5. Модифицированная задача о покрытии. Список модификаций:

1. Добавить стоимость трансляции с каждой станции
2. Добавить оценку важности каждого штата
3. Критерий выбора станции: сумма важности добавления штатов в покрытие деленная на стоимость трансляции со станции

Задание на 5\* Улучшить жадный алгоритм для данной задачи. Объяснить, почему он лучше и продемонстрировать это на тестовых данных.

Глава 9.

Задание на 4. Упражнение из главы 9.

Задание на 5. Реализовать код алгоритма динамического программирования для задачи о рюкзаке, если вес предмета рациональное число.

Задание на 5\*. Реализовать любую осмысленную прикладную задачу динамичекого программирования, где таблица трехмерная, а не двумерная.

Дополнительные главы

Решение задачи линейного программирования и 1 свой пример нелинейного программирования, где целевая функция не выпуклая.. В каждом разделе сделать свой вариант.

Задание на 4 – решение в Excel с помощью пакета «Поиск решения» (каждая задача на отдельном листе).

Задание на 5 – с помощью библиотеки для Python (или другого языка).

TBD Индивидуальные задания.