10:40 Поиск максимума — введение псевдокода

Сначала написан алгоритм поиска максимума

Потом общее описание алгоритма

NB: алгоритм в том же массиве проще, чем построение нового отсортированного массива)

Общее описание алгоритма вставкой, включающее в себя алгоритм поиска максимума

NB: все массивы нумеруются с нуля, вводится термин индекса массива

10:55 Постановка задачи сортировки (не было сделано явно)

11:13 Разбор сложности алгоритма

Около каждой строки пишем количество операций

Разбор набора операций

Оцениваем время доступа к элементам массива

Сравнительные времена выполнения операций

Резюме: все операции выполняются за одно и то же время (почему: учет времени отдельных операций требует знания устройства компьютера и ОС)

NB: не обсуждалось: в каком смысле можно говорить о времени выполнения алгоритма, не учитывая особенности компьтера? Можно говорить о зависимости времени выполнения от длины массива (до константного множителя) — ввести понятие О(n)

Оценка сложности алгоритма вставки:

Цикл поиска максимума = 5n+1

Общее время суммирования прогрессии (5/2n^2 + 7/2n - 6)

Общее время работы алгоритма. О(n^2)

NB: worst case scenario

привести пример на wsc

Не было сделано: пример на минимальное время работы алгоритма.

Наблюдение: хуже всего алг. работает, когда массив уже упорядочен

NB: для алгоритма хорошо, когда время работы в хорошем случае меньше времени работы в плохом случае

NB: проверить, не является ли массив уже удовлетворяющим нужным требованиям, можно чаще всего быстрее, чем выполнять алгоритм

Хорошо сначала выполнить вышеуказанную проверку, а потом уже запустить алгоритм.

11:30 (МР) попытка оптимизации нашего алгоритма

Воспользуемся идеей с монетками — предположим, что у нас есть два *уже* упорядоченных массива, которые надо объединить в один упорядоченный

Не было сказано (важное): первым делом, когда пишешь алгоритм, надо обозначить «дано» и «надо».

NB: all x in A, y in B: x < y

Время работы в данном случае — 1 сравнение + n переписываний

NB: из переписывания можно сделать шоу — сколько операций занимает переписывание? 1? А если подумать?

Структуры данных: списки. Время работы в случае списка время на переписывание — константа

Переписывание не нужно

NB: все было сказано очень быстро (специально) — отдельно надо рассмотреть списки и двусвязные списки, подобрать задачи. Ввести понятие набора операций, реализуемых на структуре данных; время выполнения операций на наборе операции

Оценить время работы алгоритма, который сцепляет первый и последний элемент, если выполнено условие на массивы (см. выше)

Ответ: О(n^2)

NB: алгоритм сравнения 1 с последним описан не был

Предполагалось, что каждый следующий элемент мы тащим от конца к началу — полезно иметь это на доске/слайде

Как можно улучшить этот алгоритм?

NB: дети зависли

Ясно, что надо сравнивать крайние элементы массива — лучше, сравнивать первый с первым

NB: подумать что будет, если сравнивать последний с последним

Лирическое отступление — выбор пути решения (стоит оценивать какие вообще существуют пути решения задачи)

К 10.08:

Повторение: сделать хендауты выверенные, раздать перед началом

Алгоритм мерчсорт — оценка времени

(?) Идея «разделяй и властвуй» - альтернативные пути

(?) Алгоритм радиксорт — оценка времени его работы

(?) Квадратичный алгоритм сортировки и различные эвристики

(?) Списки, придумать задачи

Темы для изложения в курсе:

Тема 1. поиск совпадений заданной длины в двух текстакх — вариации: неточные совпадения, затравки

Тема 2. pattern matching — заданное слово; множество слов.

Способы задания набора слов: списки; окрестность данного слова; профиль; реглярные выражения

Тема 3. графы, динамическое программирование.

Сначала nlogn, потом на маленьких n - O(n^2)