

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

Факультет физики

ОАД

«Билеты к экзамену»

Лектор: Корнилов М. В.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Москва
2021

Содержание

Замечания и благодарности	2
Структуры данных	3
Билет 1-3. О-нотация. Приведите примеры алгоритмов над структурами данных с кон- стантным и квадратичным (с линейным и логарифмическим) по числу элементов временем работы. Приведите примеры, в которых выбор асимптотически более медленного с точки зрения О-нотации алгоритма предпочтителен.	3
Билет 4. Сортировка. Приведите примеры методов сортировки.	4
Билет 1. Задачи оптимизации. Их классификация. Примеры	4

Замечания и благодарности

Данный конспект написан студентами и для студентов. Он может содержать опечатки, неточности и серьёзные смысловые ошибки. Над ним работали:

- Написание:

М. Блуменау

Ссылка на записи лекций

Ссылка на репозиторий с исходными файлами

Структуры данных

Билет 1-3. О-нотация. Приведите примеры алгоритмов над структурами данных с константным и квадратичным (с линейным и логарифмическим) по числу элементов временем работы. Приведите примеры, в которых выбор асимптотически более медленного с точки зрения О-нотации алгоритма предпочтителен.

В информатике временная сложность алгоритма определяется как функция от длины строки, представляющей входные данные, равная времени работы алгоритма на данном входе. Временная сложность алгоритма обычно выражается с использованием нотации «О» большое, которая учитывает только слагаемое самого высокого порядка, а также не учитывает константные множители, то есть коэффициенты. Если сложность выражена таким способом, говорят об асимптотическом описании временной сложности, то есть при стремлении размера входа к бесконечности. Временная сложность обычно оценивается путём подсчёта числа элементарных операций, осуществляемых алгоритмом. Время исполнения одной такой операции при этом берётся константой, то есть асимптотически оценивается как $O(1)$. В таких обозначениях полное время исполнения и число элементарных операций, выполненных алгоритмом, отличаются максимум на постоянный множитель, который не учитывается в О-нотации.

Примеры алгоритмов:

- $O(1)$, константный:

Определение чётности целого числа (представленного в двоичном виде)

- $O(n^2)$, квадратичный:

Сортировка пузырьком (парное сравнение соседних элементов)

Сортировка вставками (рассматриваем по элементу по одному, каждый новый элемент - в подходящее место)

- $O(n)$, линейный:

Поиск наименьшего или наибольшего элемента в неотсортированном массиве

- $O(\log(n))$, логарифмический:

Бинарный поиск (метод деления пополам)

Пример, когда асимптотически более медленный алгоритм предпочтителен:

Сложность вычисления произведения матриц по определению составляет $O(n^3)$, однако существуют более эффективные алгоритмы, применяющиеся для больших матриц.

Первый алгоритм быстрого умножения больших матриц был разработан Фолькером Штрассеном в 1969. В основе алгоритма лежит рекурсивное разбиение матриц на блоки 2×2 . Штрассен доказал, что такие матрицы можно некоммутативно перемножить с помощью семи умножений, поэтому на каждом этапе рекурсии выполняется семь умножений вместо восьми. В результате асимптотическая сложность этого алгоритма составляет $O(n^{\log_2 7}) \approx O(n^{2.81})$. Недостатком данного метода является большая сложность программирования по сравнению со стандартным алгоритмом, слабая численная устойчивость и большой объём используемой памяти. Разработан ряд алгоритмов на основе метода Штрассена, которые улучшают численную устойчивость, скорость по константе и другие его характеристики. Тем не менее, в силу простоты алгоритм Штрассена остаётся одним из практических алгоритмов умножения больших матриц.

В дальнейшем оценки скорости умножения больших матриц многократно улучшались. Однако эти алгоритмы носили теоретический, в основном приближённый характер. В силу неустойчивости алгоритмов приближённого умножения в настоящее время они не используются на практике. Лучший на данный момент - $O(n^{2.37})$.

Таким образом, можно сказать, что такие примеры в основном связаны с используемой памятью и устойчивостью.

Билет 4. Сортировка. Приведите примеры методов сортировки.

Оптимизация

Билет 1. Задачи оптимизации. Их классификация. Примеры