Дисциплина: «Теория алгоритмов»

**Лабораторная работа № 1**

# Анализ сложности наилучшего, наихудшего и среднего случая алгоритмов

### Цель работы

Получение навыков программирования алгоритмов по их описанию, оценки трудоемкости алгоритмов, выявление зависимости трудоемкости алгоритмов от объемов выборки.

### Краткие теоретические сведения

Разработка машинно-независимых алгоритмов основывается на гипотетическом компьютере, называющемся машиной с произвольным доступом к памяти (Random Access Machine) или RAM-машиной. Согласно этой модели компьютер работает таким образом:

1) для исполнения любой простой операции (+, \*, -, =, if) требуется ровно один временной шаг

2) циклы и подпрограммы не считаются простыми операциями, а состоят из несокльких простых операций. Нет смысла считать подпрограмму сортировки одношаговой операцией, т. к. для сортировки 1000000 элементов потребуется определенно намного больше времени, чем для сортировки десяти элементов. Время исполнения цикла или подпрограммы зависит от количества итераций или специфического характера подпрограммы

3) каждое обращение к памяти занимает один временной шаг. Кроме это RAM-машина обладает неограниченным объемом оперативной памяти. Кэш и диск в модели не применяются.

Время, которое тратит алгоритм как функция от размера задачи n, называют временной сложностью этого алгоритма *T*(*n*).

С помощью RAM-модели можно подсчитать количество шагов, требуемых алгоритму для исполнения любого экземпляра задачи. Но чтобы получить общее представление о том, насколько хорошим или плохим является алгоритм, нужно знать как он работает со всеми экземплярами задачи.

Сложность алгоритма в наихудшем случае – это функция, определяемая максимальным количеством шагов, требуемых для обработки любого входного экземпляра размером n.

Сложность алгоритма в наилучшем случае – это функция, определяемая минимальным количеством шагов, требуемых для обработки любого входного экземпляра размером n.

Сложность алгоритма в среднем случае – это функция, определяемая средним количеством шагов, требуемых для обработки любого входного экземпляра размером n.

Функции временной сложности трудны для работы, поэтому что перед началом работы их нужно упростить, для этой цели используются асимптотические выражения. В частности «О-большое».

В асимптотических обозначения игнорируется разница между постоянными множителями.

Формальные определения, связанные с асимптотическими обозначениями, выглядят таким образом:

1. , означает, что функция f(n) ограничена сверху функцией c\*g(n). Иными словами, существует такая константа с, для которой f(n)<=c\*g(n) при n>=n0
2. , означает, что функция f(n) ограничена снизу функцией c\*g(n). Иными словами, существует такая константа с, для которой f(n)>=c\*g(n) при n>=n0
3. , означает, что функция f(n) ограничена сверху функцией c1\*g(n), а снизу функцией c2\*g(n) (для всех n>=n0). Иными словами, существуют такие константы с1 и с2, для которых f(n)<=c1\*g(n) и f(n)>=c2\*g(n) при n>=n0.

**Порядок выполнения лабораторной работы:**

1. Выполните задания в соответствии со своим вариантом.
2. Оформите отчет по лабораторной работе.
3. Ответьте на контрольные вопросы и защитите работу.

**Требования к оформлению лабораторной работы и отчета:**

1. Работа выполняется в форме консольного приложения, в использованием функций
2. **Модуль программы** должен предваряться комментарием с текстом исходной задачи, имена переменных и компонентов должны быть понятны стороннему пользователю или сопровождаться комментарием, объясняющим их смысл. Имена проекта и модуля должны иметь осмысленные названия.
3. **Отчет по лабораторной работе** должен содержать: титульный лист, текст задания, блок-схемы алгоритмов основных алгоритмов программы (п.1), листинг самой программы с комментариями, вывод о том, какой из алгоритмов оказался более эффективным.

**Контрольные вопросы:**

1. Какое значение возвращает следующая функция? Ответ должен быть в форме функции числа n. Найдите время выполнения в наихудшем случае, используя обозначения O-большое.

function mystery(n)

r:=0

for i:=1 to n-1 do

for j:=i + 1 to n do

for k:=1 to j do

r: = r+1

return (r)

1. Сколько нужно вопросов, чтобы наверняка отгадать (целое) число от 1 до 10 100 ,используя двоичный поиск?
2. Сколько вопросов понадобится для отгадывания числа от 1 до 1000, если в качестве первого вопроса обязательно надо спросить «делится ли число на 3»?

**Вариант 1.**

1. Реализовать функции бинарного и простого поиска для массива данных (целые числа) с подсчетом количества итераций.

2. Написать функции генерации выборки для худшего, лучшего и среднего (например, генерация выборки случайным образом) случаев.

3. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для следующих значений num: 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.

**Вариант 2.**

1. Реализовать функции бинарного возведения в степень целого числа[[1]](#footnote-2) и простого возведения в степень для массива данных (целые числа) с подсчетом количества итераций.

2. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для следующих значений степеней num: 8, 16, 32, 64, 256,512, 1024.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее.

**Вариант 3.**

1. Реализовать функции поиска всех простых чисел на промежутке [1;num] (наивный алгоритм) и поиска всех простых чисел с помощью Решета Эратосфена с подсчетом количества итераций.

2. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для следующих значений num: 8, 16, 32, 64, 256,512, 1024.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее.

**Вариант 4.**

1. Реализовать функции поиска наибольшего общего делителя (НОД) двух чисел с помощью Алгоритма Евклида, алгоритма поиска НОД с бинарным сдвигом, нерекурсивного алгоритма определения НОД с подсчетом количества итераций.

2. Написать функции генерации чисел для худшего, лучшего и среднего случаев.

3. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для чисел из п.2.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Тэ3 | Т1 | Т2 | Т3 | ТЭ1 / Т1 | ТЭ2 / Т2 | ТЭ3 / Т3 |
| … | … | … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее.

**Вариант 5.**

1. Реализовать функцию умножения числа x\*y методом: число x нужно написать y раз подряд и посчитать сумму, т. е. 5\*4 = 5 + 5+ 5+ 5 = 20.

2. Реализовать функцию умножения числа x\*y поразрядным методом: т.е. 127\*211 = 127\*1+127\*10+127\*200 = 26397.

3. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1,2 для num-значных чисел: 3, 4, 5, 6, 7, 8, 9, 10.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее.

**Вариант 6.**

1. Реализовать функции бинарного поиска и бинарного поиска с рекурсией для массива данных (строки из трех символов) с подсчетом количества итераций.

2. Написать функции генерации выборки для худшего, лучшего и среднего (например, генерация выборки случайным образом) случаев.

3. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для следующих значений num: 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.

**Вариант 7 и Вариант 8**

1. Реализовать функцию для подсчета задачи 2мя способами (каждый вариант своим): Один очень ответственный студент в ночь перед экзаменом написал шпаргалку на 1 страницу А4. И теперь до экзамена он хочет успеть ее раскопировать и поделиться с друзьями по потоку. В его распоряжении находится 2 копира, один копирует страницу за x секунд, второй — за y секунд. Разрешается использовать как один копир, так и оба одновременно. Можно копировать не только с оригинала, но и с копии. Помогите студенту выяснить какое минимальное время потребуется для копирования.

1 ≤ x, y ≤ 10

2. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для следующих значений num: 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

Оцените асимптотическую сложность реализованных алгоритмов.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.

**Вариант 9 и Вариант 10**

1. Реализовать функцию для подсчета задачи 2мя способами (каждый вариант своим):

Некоторые коллекционируют марки, некоторые открытки или магнитики, а я коллекционирую сахар. У меня коллекции уже num уникальных пакетиков. Я хочу прикрепить все эти пакетики к доске и повесить на стену, доска должна быть квадратной.

Каждый пакетик сахара имеет стандартный размер длиной — 5 см и шириной 1 см. Сахар запрещается поворачивать на 90 градусов. Пакетики не должны накладываться и перекрывать друг друга. Требуется написать программу, которая вычислит минимальный размер стороны доски, которая мне потребуется для размещения всех моих пакетиков.

2. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для следующих значений num: 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

Оцените асимптотическую сложность реализованных алгоритмов.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.

**Вариант 11 и Вариант 12**

1. Реализовать функцию для решения задачи 2мя способами (каждый вариант своим):

Молодой Хакер Петр получил доступ к системе «Деканат» и теперь хочет заменить все свои минимальные оценки на максимальные. Напишите программу, которая заменяет все num оценок Василия, но наоборот (максимальные на минимальные).

Оценки ведутся по стобалльной системе.

2. Написать функции генерации выборки для худшего, лучшего и среднего (например, генерация выборки случайным образом) случаев.

3. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для следующих значений num: 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

Оцените асимптотическую сложность реализованных алгоритмов.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.

**Вариант 13.**

1. Реализовать функцию, которая на вход получает натуральное четное число и возвращает два простых числа, которые в сумме составляют входное число (с подсчетом количества итераций).

2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующих значений : 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):,

|  |  |  |  |
| --- | --- | --- | --- |
| num | Тэ1 | Т1 | ТЭ1 / Т1 |
| 8 | … | … | … |
| 16 | … | … | … |
| … | … | … | … |

По результатам сделайте вывод какой асимптотической сложности получился реализованный алгоритм.

**Вариант 14 и Вариант 15**

1. Реализовать функцию для решения задачи 2мя способами (каждый вариант своим):

Друзья решили подшутить над Василием и переставили в его клавиатуре n кнопок местами. Сколько сравнений необходимо выполнить Василию, чтобы найти все перестановки.

2. Написать функцию, которая оценивает трудоемкость алгоритмов из п. 1 для следующего количества кнопок: 5, 7, 8 , 9 для худшего, лучшего и среднего (например, генерация выборки случайным образом) случаев.

Оцените асимптотическую сложность реализованных алгоритмов.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.

**Вариант 16**

1. Реализовать функцию для сложения двух длинных чисел.

Для хранения длинного числа используется массив или вектор, в котором хранятся цифры числа. Цифры хранятся в массиве(векторе) в таком порядке, что сначала идут наименее значимые цифры (т.е. единицы, десятки, сотни, и т.д.).

2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующих оснований числа: 8, 32, 64, 88, 108 (т. е. длинное число содержит 8, 32, 64 и т. д. цифр).

Оцените асимптотическую сложность реализованного алгоритма.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |
| --- | --- | --- | --- |
| num | Тэ1 | Т1 | ТЭ1 / Т1 |
| 8 | … | … | … |
| 16 | … | … | … |
| … | … | … | … |

**Вариант 17**

1. Реализовать функцию для решения задачи:

Есть бесконечный резервуар с водой и два пустых сосуда объёмом V1 и V2 литров. Можно наливать воду из резервуара в любой сосуд до его заполнения, переливать воду из —одного сосуда в другой до заполнения второго или опустошения первого (смотря что будет раньше) и выливать воду из сосуда на землю до полного опустошения сосуда. Как таким образом можно отмерить V литра? И за сколько шагов?

При условии, что1≤𝑉1,𝑉2≤N и 0≤𝑉≤max(𝑉1,𝑉2).

Функция должна подсчитывать количество действий в решении.

Выводить строки, описывающие действия в решении. Для каждого действия это два числа:

* если это действие— переливание из одного сосуда в другой, то первое число должно быть номером сосуда, откуда надо переливать воду, а второе— номером сосуда, куда переливать;
* если это действие— набор воды из резервуара, то первое число должно быть нулём, а второе — номером сосуда, куда наливать;
* если это действие— выливание воды “на землю”, то первое число должно быть номером сосуда, а второе— нулём.

После выполнения всех операций хотя бы в одном сосуде должна находиться вода в объёме 𝑉.

Если решений не существует, выведите одно число -1.

2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующих N: 8, 32, 564, 1688, 32767 .

Оцените асимптотическую сложность реализованного алгоритма.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |
| --- | --- | --- | --- |
| num | Тэ1 | Т1 | ТЭ1 / Т1 |
| 8 | … | … | … |
| 16 | … | … | … |
| … | … | … | … |

**Вариант 18**

1. Реализовать функцию для умножения матриц A (размером x на y) и матрицы B (размером y на z) за время O(xyz).

2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующего y: 8, 32, 64, 88, 108.

Оцените асимптотическую сложность алгоритма для вариантов вложенных циклов: xyz, yxz, xzy.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Тэ3 | Т1 | Т2 | T3 | ТЭ1 / Т1 | ТЭ2 / Т2 | ТЭ3 / Т3 |
| 8 | … | … | … | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … | … | … |

По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.

**Вариант 19**

1. Реализовать функцию для умножения матриц A (размером x на y) и матрицы B (размером y на z) за время O(xyz).

2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующего y: 8, 32, 64, 88, 108.

Оцените асимптотическую сложность алгоритма для вариантов вложенных циклов: xyz, yxz, xzy.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Тэ3 | Т1 | Т2 | T3 | ТЭ1 / Т1 | ТЭ2 / Т2 | ТЭ3 / Т3 |
| 8 | … | … | … | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … | … | … |

**По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.**

**Вариант 20**

1. Реализовать функцию для нахождения в строке t из n символов, строки-образца в m символов.

2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующих m: 2, 3, 4, 5 ,6. n>100

Оцените асимптотическую сложность реализованного алгоритма.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |
| --- | --- | --- | --- |
| num | Тэ1 | Т1 | ТЭ1 / Т1 |
| 2 | … | … | … |
| 3 | … | … | … |
| … | … | … | … |

**Вариант 21 и Вариант 22**

1. Реализовать функцию для решения задачи 2мя способами (каждый вариант своим):

Полоска бумаги имеет размеры *A*×*B*. Каждый раз от нее отрезается квадрат максимального размера до тех пор, пока не получится квадрат. Сколько квадратов получится?

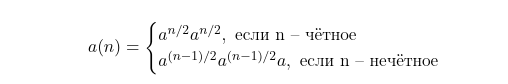
2. Написать функцию, которая оценивает трудоемкость алгоритма из п. 1 для следующих размеров 1 ≤ *A*, *B*: 8, 32, 64, 88, 108 .

Оцените асимптотическую сложность реализованных алгоритмов.

Выведите на экран значения в соответствии с таблицей (графически оформлять необязательно):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| num | Тэ1 | Тэ2 | Т1 | Т2 | ТЭ1 / Т1 | ТЭ2 / Т2 |
| 8 | … | … | … | … | … | … |
| 16 | … | … | … | … | … | … |
| … | … | … | … | … | … | … |

**По результатам экспериментов сделайте вывод о том, какой из алгоритмов оказался эффективнее и для каких размеров данных.**

1.  [↑](#footnote-ref-2)