

О.Ю. Жарова

***Изучение понятия псевдокод.
Определение наилучшего и наихудшего случая алгоритма***

Методические указания по выполнению лабораторной работы
по дисциплине «Алгоритмы и структуры данных»

УДК 681.3-7

ББК 32.973

Данные методические указания разработаны в соответствии с учебным планом специальности 090903.65 «Информационная безопасность автоматизированных систем».

Указания рассмотрены и одобрены:
кафедрой ЭИУ6-КФ «Информационная безопасность автоматизированных систем»

протокол № _____ от _____ 20__ г.

Зав. кафедрой ЭИУ6-КФ _____ к.т.н., доц. Мазин А.В.

методической комиссией факультета ЭИУК

протокол № _____ от _____ 20__ г.

Председатель методической комиссии

факультета ЭИУК _____ к.т.н., доц. Адкин М.Ю.

методической комиссией Калужского филиала МГТУ им. Н.Э. Баумана

протокол № _____ от _____ 20__ г.

Председатель методической комиссии

КФ МГТУ им. Н.Э. Баумана _____ Зам. директора по учебной работе Перерва О.Л.

Рецензент:

к.т.н., доц. кафедры ЭИУ2-КФ _____

Автор

ассистент кафедры ЭИУ6-КФ

Жарова Ольга Юрьевна

В методических указаниях изложены принципы составления алгоритмов с использованием различных способов записи. Предназначено для студентов специальности 090903.65 «Информационная безопасность автоматизированных систем» и может быть рекомендовано к применению при проведении лабораторных работ по курсу «Алгоритмы и структуры данных».

© Калужский филиал МГТУ им. Н.Э. Баумана, 20__ г.

© Кафедра ЭИУ6-КФ, 20__ г.

© Жарова О.Ю., 20__ г.

СОДЕРЖАНИЕ

Введение.....	4
1 Теоретические сведения.....	5
1.1 Способы записи алгоритмов.....	5
1.2 Псевдокод.....	6
1.3 Наилучший и наихудший случай.....	7
2 Практическая часть.....	7
2.1 Пример составления алгоритма.....	7
2.2 Выполнение лабораторной работы.....	9
2.3. Варианты индивидуальных заданий.....	10
2.4 Требования к оформлению отчета.....	11
3 Контрольные вопросы.....	11
Литература	11

Введение

Тема работы: Изучение понятия псевдокод. Определение наилучшего и наихудшего случая алгоритма.

Требования к оборудованию:

- Компьютер уровня не ниже, чем Pentium CoreDuo/RAM 1024 MB/HDD 1 GB;
- Операционная система Windows версии XP и выше с типовым вариантом установки;
- среда программирования Visual Studio8 и выше.

Цель работы: выработать практические навыки составления алгоритмов с использованием псевдокода.

Содержание работы: изучить основные способы записи алгоритмов, освоить псевдокод, составить алгоритм с его использованием для решения индивидуальной задачи.

Требования к отчету. Индивидуальный отчет студента должен быть представлен в электронной форме с типовым титульным листом и должен содержать:

1. Постановку задачи.
2. Описание выполнения работы.
3. Экранные формы хода работы.

Длительность работы: 4 академических часа.

Защита работы: собеседование с преподавателем по контрольным вопросам.

1 Теоретические сведения

1.1. Способы записи алгоритмов

Алгоритм - это совокупность действий, необходимых для решения задачи. Алгоритм отличается от программы тем, что в нем содержится только описание действий, производимых над данными, но полностью отсутствуют какие-либо описания данных. Алгоритмы содержат определение пошагового процесса обработки данных с описанием преобразований данных и функций управления.

Существуют пять наиболее употребительных способов записи алгоритма: словесный, формульно-словесный, графический, при помощи псевдокодов и языков программирования (последний способ не будет рассматриваться в данной работе).

Словесное задание описывает алгоритм-инструкцию о выполнении действий в определенной последовательности с помощью слов и предложений естественного языка. Форма изложения произвольна и устанавливается разработчиком.

В **формульно-словесном** способе записи инструкция о действиях содержит формальные символы и выражения (формулы) в сочетании со словесными пояснениями.

Графическая запись – это изображение алгоритма с помощью геометрических фигур, называемых блоками.

Наряду с графическим способом представления широко используется **псевдокод**. Псевдокодом называется система правил записи алгоритма с использованием набора определенных конструкций для описания управляющих действий.

Псевдокод позволяет формально изображать логику алгоритма, используя стандартизированные конструкции естественного языка для изображения управления и сохраняя возможности языка для описания действий по обработке информации. Данный способ тесно связан со структурным подходом к программированию. Псевдокод занимает промежуточное положение между естественным языком и языком программирования. Его применяют преимущественно для того, чтобы подробнее объяснить работу программы, что облегчает проверку правильности программы. Кроме того, псевдокод дает программисту большую свободу в изображении алгоритма. Требуется только употреблять стандартные управляющие конструкции и правила записи.

1.2 Псевдокод

Псевдокодом называется подробное описание алгоритма на структурированном и частично формализованном подмножестве английского языка.

Основные правила, задающие псевдокод:

- действия описываются предложениями, помещаемыми в отдельной строке;
- последовательность действий описывается аналогичной последовательностью описывающих их предложений;
- разветвление описывается конструкцией IF-THEN-ELSE;
- завершение разветвления описывается конструкцией END-IF;
- выбор одного из нескольких направлений описывается конструкцией CASE;
- завершение структуры выбора описывается конструкцией END-CASE;
- цикл описывается конструкцией DO-WHILE;
- завершение цикла описывается конструкцией END-DO.

Между словами IF-THEN указывают проверяемое условие, между THEN-ELSE – действие, выполняемое при истинности условия (линия TRUE), между ELSE и END-IF – действие при ложности условия (линия FALSE).

Во вложенных конструкциях разветвления для внешнего IF слово ELSE можно опустить, если действий в этой ветви не предусматривается. Для вложенных IF пустая конструкция ELSE обязательна.

После слова CASE записывается переменная – селектор выбора направления. Затем между селектором и конструкцией END-CASE записывают все возможные значения селектора и соответствующие им действия.

При использовании цикла FOR применяется следующая конструкция:

FOR счетчик цикла = <начальное значение> TO <конечное значение> DO

<Тело цикла>

END-DO

После слов DO-WHILE записывают условие повторяемости действий области цикла, находящейся между условием и словами END-DO.

Для получения наглядности алгоритма в псевдокодах введен ряд

правил его изображения:

- служебные слова следует располагать друг под другом и в рамках одной конструкции записывать с одной позиции;
- условия следует записывать в той же строке, что и соответствующие им служебные слова;
- действия записывают на отдельных строках со сдвигом вправо относительно служебной конструкции;
- вложенные конструкции записывают со сдвигом вправо относительно объемлющих конструкций.

1.3. Наилучший и наихудший случай

Наилучшим случаем для алгоритма является такой набор данных, на котором алгоритм выполняется за минимальное число операций. Такой набор данных представляет собой комбинацию значений, на котором алгоритм выполняет меньше всего действий. Если мы исследуем алгоритм поиска, то набор данных является наилучшим если искомое значение (обычно называемое целевым значением или ключом) записано в первой проверяемой алгоритмом ячейке. Такому алгоритму, вне зависимости от его сложности потребуется одно сравнение. При поиске в списке, вне зависимости от его длины, наилучший случай требует постоянного времени.

Наихудший случай представляет собой набор входных данных, на котором алгоритм выполняет максимальное количество действий. Для алгоритма поиска подобные входные данные — это список, в котором искомый элемент находится последним из рассматриваемых или вообще отсутствует.

2 Практическая часть

2.1. Пример составления алгоритма

В качестве иллюстрации приведем алгоритм вычисления функции:

$$y = \begin{cases} \prod_{i=1}^n \sin ix & \text{при } x > b, \\ \sum_{i=1}^n \cos ix & \text{при } [a, b], \\ e^x & \text{при } x < a. \end{cases}$$

сначала графическим способом (Рис.1.), а затем на псевдокоде.

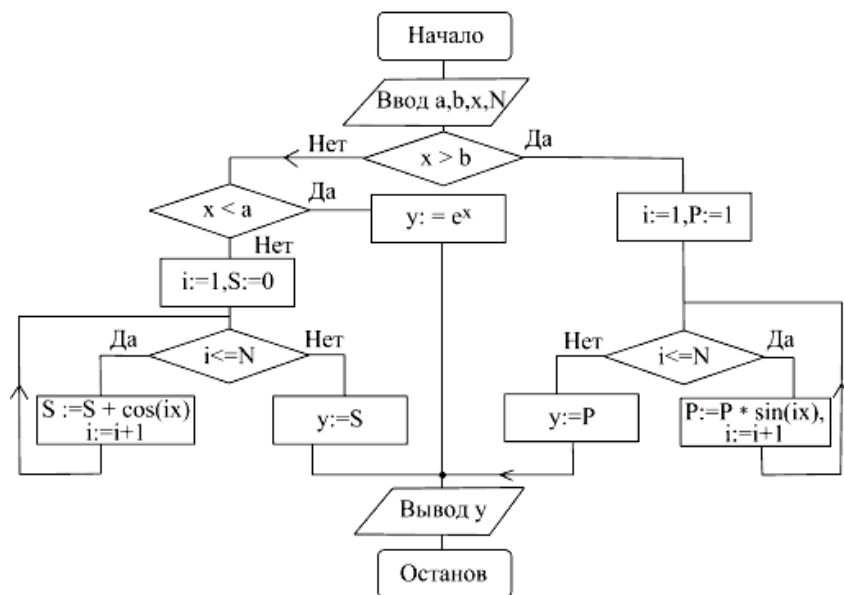


Рис.1. Алгоритм вычисления функции

Этот же алгоритм с помощью псевдокода:

Ввести границы интервала, число членов ряда и значение аргумента

IF значение аргумента больше верхней границы интервала:

THEN

Задать начальное значение счетчика циклов

Задать начальное значение произведения членов ряда

DO-WHILE значение счетчика меньше числа членов ряда

Вычислить по рекуррентной формуле произведение членов
ряда
Инкремент счетчика циклов

END-DO

Вывести заголовок «Произведение членов ряда»

Перейти к выводу результата

ELSE

IF значение аргумента меньше нижней границы интервала

THEN

Вычислить экспоненту от заданного аргумента

Вывести заголовок «Экспонента аргумента»

Перейти к выводу результата

ELSE

Задать начальное значение счетчика цикла

Задать начальное значение суммы членов ряда

DO-WHILE значение счетчика меньше числа членов ряда

Вычислить по рекуррентной формуле сумму членов
ряда

Инкремент счетчика циклов

END-DO

Вывести заголовок «Сумма членов ряда»

Перейти к выводу результата

END-IF нижняя граница интервала

END-IF верхняя граница интервала

Вывод результата

Основное – использование только служебных конструкций и соблюдение правил ступенчатой записи.

2.2. Выполнение лабораторной работы

1. Необходимо составить алгоритм согласно индивидуальному заданию, используя псевдокод.
2. Проанализировать полученный алгоритм, найти набор входных данных для наихудшего и наилучшего случая.
3. Реализовать алгоритм на любом выбранном языке программирования.
4. Составить отчет в электронном виде.
5. Ответить на дополнительные вопросы преподавателя.

2.3. Варианты индивидуальных заданий

1. Напишите в псевдокоде алгоритм подсчитывающий количество прописных букв в текстовом файле. Сколько сравнений требуется алгоритму? Каково максимальное и минимальное возможное значение числа операций увеличения счетчика? Выразите ответ через число N символов во входном файле.

2. Напишите в псевдокоде алгоритм подсчитывающий количество знаков препинания в текстовом файле. Сколько сравнений требуется алгоритму? Каково максимальное и минимальное возможное значение числа операций увеличения счетчика? Выразите ответ через число N символов во входном файл.

3. Напишите алгоритм определяющий наименьшее число из трех не используя временную переменную. Сколько сравнений требуется алгоритму. Какой случай является наилучшим и наихудшим?

4. Напишите алгоритм поиска наименьшего числа в списке. Сколько сравнений требуется алгоритму. Определите наилучший и наихудший случай.

5. Напишите алгоритм поиска числа вдвое большего наименьшего числа списка если оно есть в файле из неизвестного количества элементов. Определите количество операций при наилучшем и наихудшем случае.

6. Напишите алгоритм поиска всех чисел больше 100 с последующим вычислением их среднего значения. Сколько сравнений требуется алгоритму. Какой случай является наилучшим и наихудшим?

7. В файле записано некоторое неизвестное количество чисел. Написать в псевдокоде алгоритм для подсчета среднего значения чисел в файле. Каково максимальное и минимальное возможное значение числа операций? Выразите ответ через число N чисел во входном файле.

8. Напишите алгоритм для поиска второго по величине элемента в списке из N значений. Сколько сравнений делает алгоритм в наихудшем и наилучшем случае?

9. Напишите алгоритм, который по данному списку чисел и среднему значению этих чисел определяет, превышает ли число элементов списка, больших среднего значения, число элементов списка, меньших среднего значений или наоборот. Опишите группы на которые распадаются возможные наборы выходных данных. Какой случай является наилучшим и наихудшим?

10. Написать алгоритм определяющий наличие в произвольном

списке элемента в трое больше заданного, и выдающий индекс элемента при положительном результате поиска. Каково максимальное и минимальное возможное значение числа операций?

11. В файле записано некоторое количество слов. Напишите в псевдокоде алгоритм поиска слов с количеством символов больше заданного числа, посчитать их количество. Каково максимальное и минимальное возможное значение числа операций?

12. Дана произвольная последовательность символов, определить какое количество из них является числами, и больше или меньше это количество заданного числа N . Каково максимальное и минимальное возможное значение числа операций?

2.4. Требования к оформлению отчета

Отчет по лабораторной работе должен содержать:

1. Титульный лист установленного образца.
2. Постановку задачи.
3. Описание хода выполнения работы (псевдокод алгоритма, текст программы на любом выбранном языке программирования, результаты работы программы).
4. Вывод.

3 Контрольные вопросы

- 1) Что такое псевдокод?
- 2) Отличие псевдокода от словесно-формульного задания алгоритма?
- 3) Что такое наилучший случай?
- 4) Какой набор данных является наихудшим случаем для полученного алгоритма?

Литература

1. Дж. Макконелл. Основы современных алгоритмов. 2-е дополнительное издание. – М.: Техносфера, 2004. – 368 с.