МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Методические рекомендации к лабораторным работам для студентов направления подготовки 09.03.04 «Программная инженерия» и 09.03.01 «Информатика и вычислительная техника» очной формы обучения

Могилев 2022

УДК 510.5(076.5)(075.8)
ББК 22.12:32.97
Л69

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления» «11» января 2022 г., протокол № 5

Составитель ст. преподаватель И. А. Беккер

Рецензент канд. техн. наук, доц. С. К. Крутолевич

Методические рекомендации к лабораторным работам содержат краткие теоретические сведения, задания, контрольные вопросы. Предназначены для студентов направления подготовки 09.03.04 «Программная инженерия» и 09.03.01 «Информатика и вычислительная техника» очной формы обучения.

Учебно-методическое издание

ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Ответственный за выпуск	А. И. Якимов
Корректор	А. А. Подошевко
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс. Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2022

Содержание

Введение

Дисциплина «Логика и теория алгоритмов» относится к математическому и естественно-научному циклу учебного плана и является важным инструментом при изучении различных дисциплин специальности. В результате изучения дисциплины у студентов формируются математическая культура, знание законов и символических обозначений математической логики, умения строить логические рассуждения, доказательства, понимание основных концепций и принципов теории алгоритмов.

Методические рекомендации к лабораторным работам являются неотъемлемой частью комплексного методического обеспечения дисциплины и способствуют развитию умения логически верно, аргументированно и ясно строить устную и письменную речь, формированию готовности использовать основные законы математической логики при формализации рассуждений, способности применять методы теоретического и экспериментального исследования алгоритмов и оценивать временную и емкостную сложность программного обеспечения.

Процесс выполнения и защиты лабораторной работы студентом состоит в том, что следует успешно пройти следующие этапы учебной деятельности:

– выполнить согласно варианту, выданному преподавателем, задание;

– составить отчет с описанием хода работы и результатов, сделать вывод по заданию (требования к отчету представлены в приложении А);

– предъявить разработанное программное обеспечение с тестированием его на различных наборах входных данных (лабораторные работы № 6–10);

– продемонстрировать знание теоретического материала применительно к заданию лабораторной работы (знать все термины, свойства, закономерности, встречающиеся в работе; ответить на контрольные вопросы, приведенные в конце лабораторной работы).

Контрольные вопросы приводятся последовательно, отвечать на них следует по порядку, ответы образуют связный цельный рассказ.

1 Лабораторная работа № 1. Равносильные преобразования алгебры высказываний

Цель работы: изучить законы равносильных преобразований формул; сформировать навыки решения проблемы разрешимости формул алгебры высказываний.

Теоретические сведения

Формула алгебры высказываний, везде принимающая значение ИСТИНА, называется тавтологией. В математической логике тавтологии служат для записи ее законов и образуют самый важный класс формул.

Формула алгебры высказываний, принимающая везде значение ЛОЖЬ, называется противоречием (тождественно ложной).

Формула называется выполнимой, если существует такой набор значений переменных, при котором эта формула принимает значение ИСТИНА и при этом она не является тавтологией.

Задачу «К какому классу относится формула?» называют проблемой разрешимости алгебры логики.

Эффективную процедуру решения этого вопроса дают таблицы истинности.

Процедура их построения называется разрешающей процедурой и благодаря ей логика высказываний является разрешимой.

Задание 1

Проверьте равносильность формул A и B, используя законы алгебры высказываний.

1 $A = xy \lor \neg x\neg z \lor x\neg z$, $B = x\neg y\neg z \lor \neg z$.

2 $A = x\neg z \lor xy \lor y\neg z$, $B = x\neg(yz) \lor \neg xz$.

3 $A = (x \lor y)(y \lor z)(z \lor x)$, $B = xy \lor yz \lor zx$.

4 $A = (x \lor y)(x \lor z)(y \lor \omega)(z \lor \omega)$, $B = x\omega \lor yz$.

5 $A = \neg(\neg xyz)$, $B = x \lor x\neg z \oplus \neg y$.

6 $A = xy\neg z \lor xy \lor z\neg z$, $B = \neg(\neg x \leftrightarrow \neg y)$.

7 $A = x \oplus y$, $B = \neg\,(xy \to \neg(\neg x \lor y))$.

8 $A = x(y \lor z) \lor y\neg x\neg z$, $B = x \oplus y \lor \neg z$.

9 $A = \neg x \lor y \oplus z$, $B = \neg\,(\neg\,(\neg x \lor y) \leftrightarrow z)$.

10 $A = \neg(x \lor y) \leftrightarrow \neg(\neg xy)$, $B = (x \lor \neg y)(\neg x \lor xy)$.

11 $A = x(x \oplus y)$, $B = \neg x\neg y \to \neg(x \lor y)$.

12 $A = x \lor y \lor z \to \neg\,(x \lor z)$, $B = x \leftrightarrow z$.

13 $A = \neg x\neg y\neg z \oplus \neg x\neg yz \lor \neg xy$, $B = \neg z \to \neg(\neg x\neg y)$.

14 $A = x \to (xy \to ((x \to y) \to y)z)$, $B = y \leftrightarrow (x \to z)$.

15 $A = \neg((x \to y) \lor ((x \to z)y))$, $B = (x\neg y)(\neg y \leftrightarrow x\neg z)$.

16 $A = ((x \oplus y) \to x \lor y)((\neg x \to y) \to (x \lor y))$, $B = \neg(xy)$.

17 $A = ((x \oplus y \to x)(\neg x \to y \to xy)),\ B = \neg(xy) \vee \neg xy \ \vee \neg yx.$

18 $A = \neg\, (\neg\, (\neg x \vee y) \to z),\ B = \neg xy\neg z \oplus \neg xy\neg z \to \neg xy.$

Задание 2

Определите, являются ли формулы тавтологиями двумя способами – через таблицу истинности и методом редукции – и сверьте результаты.

Пример исследования формулы методом редукции

Предположим, что существует набор, на котором формула принимает значение, равное ЛОЖЬ (Л):

$$((A\to B)\to B)\to B = Л.$$

Тогда $((A\to B)\to B) = И$, $B = Л$.

Из первого равенства $(A\to Л) \to Л = И$.

Если $A = И$, то $(T\to Л) \to Л = И$. Следовательно, при $A = И$, $B = Л$ формула принимает значение, равное Л. Таким образом, она не является тавтологией.

1 $(\neg A \to \neg B)\to(A \to B).$
2 $(A \to B) \to ((C \to B) \to (A\&C \to B)).$
3 $(A \to B) \to ((B \to C) \to (A \to C)).$
4 $(A \to (B \to C)) \to ((A \to B) \to (A \to C)).$
5 $((A \to B) \to A) \to ((A \to B) \to B).$
6 $(A \to C) \to (A \to B \vee C).$
7 $(\neg A \to B) \to (\neg B \to A).$
8 $((A \to B) \to (A \to C) \to (A \to (B \to C))).$
9 $(B \to C) \to (B \to (A \to C)).$
10 $(B \to A) \to (A \vee B \to A).$
11 $A \,\&\, B \to (B \to C).$
12 $(B \to A \vee C)\to((B \to C)\to((D \to C) \to B \vee D \to C)).$
13 $((B \to A) \to C) \to (A \to C).$
14 $(A \,\&\, B) \vee (C \,\&\, D) \to (A \vee B) \,\&\, (C \vee D).$
15 $(A \to B) \,\&\, (C \to D) \to (A \vee C\to D \vee D).$
16 $(A \vee B \to C) \to (A \to C) \vee (B \to C).$
17 $(A \,\&\, B \to C) \to (A \to C) \to (B \to C).$
18 $(A \to B) \,\&\, (C \to D) \to (A \,\&\, C \to B \,\&\, D).$
19 $(A \to B) \to ((\neg A \to B) \to B).$
20 $(A \to (B \to C)) \to (B \to (A \to C)).$
21 $(\neg B \to A) \to ((\neg B \to A) \to B).$
22 $A \to (\neg B \to (\neg A \to B)).$
23 $\neg(A \,\&\, B) \to (A \,\&\, B \to B).$
24 $(B \to C) \to (A \vee C \to (B \to C)).$
25 $(A \to (B \to C)) \to (A \vee B \to C).$

Контрольные вопросы

1 Дайте определение логике высказываний, алгебре высказываний.
2 Определите операции над высказываниями.
3 Какие им соответствуют операции над множествами?
4 Операции над множествами опишите предикатами и диаграммами Эйлера.
5 Что такое логика высказываний, алгебра высказываний?
6 Перечислите основные законы алгебры высказываний.
7 В чем состоит проблема разрешимости алгебры логики; формулы алгебры логики? Какова процедура решения этой проблемы?
8 Какие выделяют классы формул логики высказываний?
9 Какие формулы называются равносильными?
10 Сколько строк в таблице истинности?

2 Лабораторная работа № 2. Решение логических задач и логических уравнений методами алгебры логики

Цель работы: изучить методы исследования отношений между высказываниями, правила построения логического вывода.

Теоретические сведения

При решении логических задач можно использовать методы рассуждения, построения таблицы истинности для конечного сложного высказывания, табличный метод.

Для построения из посылок новых высказываний (заключений, выводов) используют следующие основные правила логики высказываний:

- правило силлогизма: $A \to B, B \to C \vdash A \to C$;
- правило контрапозиции: $A \to B \vdash \neg B \to \neg A$.

В начале процедуры проверяют на истинность все посылки.

Если хотя бы одна из посылок ложна, то применять правила логики высказываний не имеет смысла.

Задание 1

Решите задачи с использованием математической логики.

1 Два свидетеля дают следующие показания.

Первый свидетель. Если А виновен, то В и подавно виновен, а С – невиновен.

Второй свидетель. Виновны двое. А точно виновен и виновен один из оставшихся, но кто именно сказать не могу.

Какие заключения о виновности А, В и С можно сделать на основании свидетельских показаний?

2 Сергей приехал поступать в университет и послал домой три сообщения:

«Если я сдам математику, то информатику я сдам только при условии, что не завалю диктант».

«Не может быть, чтобы я завалил и диктант, и математику».

«Достаточное условие завала по информатике – это двойка по диктанту».

После сдачи экзаменов оказалось, что только одно сообщение было ложным. Как Сергей сдал экзамены?

3 Аня, Вика и Сергей решили пойти в кино. Их друг высказал следующие предположения.

Аня пойдет в кино только тогда, когда пойдут Вика и Сергей.

Аня и Сергей пойдут в кино вместе или же оба останутся дома.

Чтобы Сергей пошел в кино, необходимо, чтобы пошла Вика.

Когда ребята пошли в кино, оказалось, что друг немного ошибся: из трех его утверждений истинными оказались только два. Кто из ребят пошел в кино?

4 В нарушении правил обмена валюты подозреваются четыре работника банка: A, B, C, D.

Известно, что:

- если A нарушил, то и B нарушил правила обмена валюты;
- если B нарушил, то и C нарушил или A не нарушал;
- если D не нарушил, то A нарушил, а C не нарушал правила;
- если D нарушил, то и A нарушил.

Задание 2

Выведите (если возможно) заключение из каждого набора посылок.

1 Если Петров играет в нашей команде, то он обязательно храбрый и хорошо владеет техникой удара. Но он не входит в состав нашей команды.

2 Если это птица, то она летает. Летающий по небу видит землю с высоты.

3 Птицы умеют летать. Бабочки – не птицы.

4 У людей две ноги. У страуса тоже две ноги.

5 В сырую погоду у Саши болит горло. Осенью сыро.

6 Все звери – это млекопитающие. Млекопитающие выкармливают детенышей молоком. Волки – это звери.

7 Днем светло. При плохом освещении бабушка не может вязать.

8 Рыбы умеют плавать. Если это морская звезда, то она тоже – рыба.

9 У самолета есть крылья. У насекомых также есть крылья.

10 Если идет дождь, то это наводит скуку. Осенью идет дождь.

11 В дождливый день Иван Петрович не ходит на прогулку. Без свежего воздуха у него пропадает аппетит.

12 Воробей – это животное. Все животные имеют позвоночник.

13 Рыбаки ловят рыбу. Тот, кто ловит рыбу, – оптимист. Оптимист не предается отчаянию.

14 Если я читал статью, то она была в интернете. Информация в интернете может быть небылицей.

15 Если человек жадный, то у него не будет друзей. Человек без друзей не имеет поддержки.

16 Если человек стремится постичь смысл жизни, то он читает книги. Боксеры не читают книги.

17 Олег или устал, или он болен. Если он устал, то он раздражен. Он не раздражен.

18 Совершивший подвиг получает награду. Если человек трусливый, то он не достоин награды.

19 Лекарства неприятны на вкус. Таблетки – это лекарства.

20 Отличник всегда много читает. У читающего человека широкий кругозор.

21 Если студент прогуляет много занятий, то он получит двойку на экзамене. Если он получит двойку на экзамене, то пропадут каникулы. Студент может хорошо отдохнуть, если у него будут каникулы.

22 В хорошую погоду кошка ходит гулять. Если кошка больна, то она сидит дома.

23 Если Кандидатов победит на выборах, он будет доволен, а если он будет доволен, то он плохой борец в предвыборной кампании.

24 Деревья, которые растут в этом саду, плодоносят. Деревья, которые плодоносят, дают хороший урожай. Деревья, дающие хороший урожай, получают тщательный уход. Ни одно дерево в этом саду не получает тщательного ухода.

Контрольные вопросы

1 Что такое логическое следование, силлогизм, дедукция, индуктивное умозаключение?

2 Как называются элементы в силлогизме?

3 Приведите правила modus ponens, modus tollens.

4 В каком случае нет смысла применять правила логики высказываний?

5 Можно ли переставлять местами посылки?

3 Лабораторная работа № 3. Нормальные формы логических функций

Цель работы: изучить методы приведения формул логики высказываний к совершенным нормальным формам.

Теоретические сведения

Дизъюнктивной нормальной формой (*ДНФ*) *формулы* называется равносильная ей формула, представляющая собой дизъюнкцию конъюнктов.

Для любой формулы алгебры логики можно получить ДНФ, причем не единственную.

Совершенной дизъюнктивной нормальной формой (*СДНФ*) *формулы А* называется такая ДНФ формулы А, которая удовлетворяет свойствам:

1) все конъюнкты различны;
2) каждый конъюнкт содержит все входные переменные;
3) ни один конъюнкт не содержит переменную и ее отрицание;
4) ни один конъюнкт не содержит одну и ту же переменную дважды.

СДНФ формулы А может быть получена с помощью таблицы истинности, а может и через равносильные преобразования формулы А:

– сначала путем равносильных преобразований формулы А получаем одну из ее ДНФ;

– если в этой ДНФ входящий в нее конъюнкт В не содержит переменную x_i, то применяем равносильность $B \,\&\, (x_i \vee \neg x_i) \equiv B$ и заменяем конъюнкт В на два конъюнкта: $B \,\&\, x_i$ и $B \,\&\, \neg x_i$;

– когда в ДНФ формулы А входят два одинаковых конъюнкта, применяем равносильность $B \vee B \equiv B$ и один из них отбрасываем;

– если конъюнкт В в ДНФ содержит переменную x_i и ее отрицание $\neg x_i$, то $B \equiv 0$ и его исключают из ДНФ;

– если конъюнкт В в ДНФ содержит переменную x_i дважды, то равносильностью $x_i \,\&\, x_i \equiv x_i$ одну переменную отбрасываем.

Конъюнктивной нормальной формой (*КНФ*) *формулы* называется равносильная ей формула, представляющая собой конъюнкцию дизъюнктов.

Для любой формулы алгебры логики можно получить КНФ, причем не единственную.

Совершенной конъюнктивной нормальной формой (*СКНФ*) *формулы А* называется такая КНФ формулы А, которая удовлетворяет свойствам:

1) все дизъюнкты различны;
2) каждый дизъюнкт содержит все входные переменные;
3) ни один дизъюнкт не содержит переменную и ее отрицание;
4) ни один дизъюнкт не содержит одну и ту же переменную дважды.

СКНФ формулы А может быть получена с помощью таблицы истинности, а может и через равносильные преобразования формулы А:

– путем равносильных преобразований формулы А получаем одну из ее КНФ;

– если в этой КНФ входящий в нее дизъюнкт В не содержит переменную x_i, то применяем равносильность $B \vee (x_i \,\&\neg x_i) \equiv B$ и заменяем дизъюнкт В на два дизъюнкта: $B \vee x_i$ и $B \vee \neg x_i$;

– когда в КНФ формулы А входят два одинаковых дизъюнкта, применяем равносильность $B \,\&\, B \equiv B$ и один из них отбрасываем;

– если конъюнкт В в КНФ содержит переменную x_i и ее отрицание $\neg x_i$, то $x_i \vee \neg x_i \equiv 1$ и В исключаем из ДНФ;

– если конъюнкт В в КНФ содержит переменную x_i дважды, то, применяя равносильность $x_i \vee x_i \equiv x_i$, одну переменную отбрасываем.

Задание 1

Представьте булевы функции в виде СДНФ, СКНФ двумя способами и сравните полученные результаты.

1 $x \to (x \to z)$.
2 $(x \lor \neg y \lor z)(\neg x \neg y \neg z)$.
3 $(\neg x \neg y \lor \neg y \neg z) \leftrightarrow (x \lor z \to y)$.
4 $xy \lor x(z \lor y)z$.
5 $(x \to (x \to y)) \to z$.
6 $(x \leftrightarrow y) \leftrightarrow z$.
7 $\neg(xy \to x) \lor (x(y \lor x))$.
8 $(\neg x \to \neg y) \to (xy \to xz)$.
9 $((x \to y) \to \neg x) \to (x \to yx)$.
10 $\neg((x \land y) \to \neg x) \neg(xy \to \neg y)$.
11 $(z \to x) \to (\neg(y \lor z) \to x)$.
12 $\neg(x(y \lor z)) \to (xy \lor z)$.
13 $((((x \to y) \to \neg x) \to \neg y) \to \neg z) \to z$.
14 $((x \to y) \to (z \to \neg x)) \to (\neg y \to \neg z)$.
15 $(x \to (y \to z)) \to ((x \to \neg z) \to (x \to \neg y))$.

Задание 2
Найдите более простой вид формул, имеющих СНФ (упростите).
1 $xy \lor x\neg y \lor \neg xy$.
2 $(x \lor \neg y)(\neg x \lor \neg y)(\neg x \lor y)$.
3 $xyz \lor \neg xyz \lor x\neg yz$.
4 $(x \lor y \lor \neg z)(\neg x \lor y \lor z)(x \lor \neg y \lor \neg z)$.
5 $(x \lor \neg y \lor z)(\neg x \lor \neg y \lor \neg z)$.
6 $xyz \lor \neg xyz \lor x\neg yz \lor \neg x\neg y\neg z$.

Задание 3
Постройте СДНФ для всякой тавтологии, содержащей:
а) две переменных;
б) три переменных;
в) четыре переменных. (Примеры приводить не надо!)

Задание 4
Найдите СКНФ для всякого противоречия, содержащего:
а) две переменных;
б) три переменных;
в) четыре переменных. (Примеры приводить не надо!)

Контрольные вопросы

1 Что такое булева функция, булева алгебра?
2 Перечислите основные законы булевой алгебры.
3 Что такое ДНФ, СДНФ? Сколько их может быть у формулы?
4 Какие есть методы построения нормальных форм формулы?
5 Как связано количество переменных формулы с количеством элементов в СНФ?

6 Дайте определение булевой функции, булевой алгебре.
7 Приведите основные свойства булевых функций (с названиями).

4 Лабораторная работа № 4. Теория предикатов первого порядка

Цель работы: научиться использовать аппарат логики предикатов и исследовать отношения между ними, находить область истинности предикатов.

Теоретические сведения

Предикатом называется функция, аргументы которой принимают значения из некоторого множества, а сама функция – значение ЛОЖЬ или ИСТИНА.

Предикат называется n-местным ($n \in N$), если соответствующая функция есть функция от n аргументов. Высказывание – предикат без аргумента (нульместный предикат).

Пример – Бинарные (2-местные) предикаты $P(x, y) = (x > y)$, $Q(x, y) =$ = «x делит y» заданы на множестве N^2.

Множество тех пар (x, y), для которых бинарный предикат принимает значение ИСТИНА, есть область истинности $I_{P(x,y)}$ предиката P(x, y). Табличную запись функции называют матрицей предиката.

Пусть R(x) и E(x) – два одноместных предиката, заданные на некотором множестве M. Определим логические операции над ними.

Конъюнкция. $P1(x) \equiv R(x) \& E(x)$ – это предикат, который истинен для тех, и только для тех объектов из M, для которых оба предиката истинны. Таким образом, область истинности предиката P1(x) равна пересечению областей истинности предикатов R(x) и E(x).

Дизъюнкция. $P2(x) \equiv R(x) \vee E(x)$ – это предикат, который ложен для тех, и только для тех объектов из M, для которых оба предиката ложны. Таким образом, область истинности предиката P2(x) равна объединению областей истинности предикатов R(x) и E(x).

Отрицание. $P3(x) \equiv \neg R(x)$ – это предикат, который истинен для тех, и только для тех объектов из M, для которых предикат R(x) ложен. Его область истинности является дополнением области истинности предиката R(x).

Задание 1

Постройте матрицы предикатов на множестве натуральных чисел. Для построенных матриц предикатов найдите значения всевозможных предикатов с квантификациями.

1 $P(x, y) = (x + y = 5)$.
2 P(x, y) = «x – простое число» & «y – чётное число».
3 $P(x, y) = (2x = y^2)$.
4 $P(x, y) = (x + y = 8)$.

5 $P(x, y) = (x/y = y/x)$.
6 $P(x, y) = (x + y = 2y)$.
7 $P(x, y)$ = «x – простое число» & «y кратно 3».
8 $P(x, y) = (x = y + 1)$.
9 $P(x, y) = (x + 2 = y)$.
10 $P(x, y) = (x + 1 = y)$.

Задание 2

Найдите область истинности предиката, заданного на множестве $M = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

1 $\neg P(x) \vee \neg Q(x)$, где $P(x)$ = «x нечетное», $Q(x)$ = «число 20 кратно числу x».

2 $P(x) \to \neg Q(x)$, где $P(x)$ = «x не больше 7», $Q(x)$ = «x делится на число 4».

3 $\neg P(x) \oplus \neg Q(x)$, где $P(x)$ = «x больше 3», $Q(x)$ = «x не превосходит число 5».

4 $\neg P(x) \to Q(x)$, где $P(x)$ = «$x - 7 > 0$», $Q(x)$ = «x делится на 3».

5 $P(x) \wedge \neg Q(x)$, где $P(x)$ = «60 кратно числу x», $Q(x)$ = «x не больше, чем 7».

6 $P(x) \to \neg Q(x)$, где $P(x)$ = «x не меньше 6», $Q(x)$ = «x четное».

7 $\neg P(x) \oplus Q(x)$, где $P(x)$ = «x делится на 3», $Q(x)$ = «x не больше 6».

8 $\neg P(x) \vee Q(x)$, где $P(x)$ = «x не меньше 6», $Q(x)$ = «96 кратно числу x».

9 $P(x) \wedge \neg Q(x)$, где $P(x)$ = «x нечетно», $Q(x)$ = «$|x|$ не больше 5».

10 $\neg P(x) \to Q(x)$, где $P(x)$ = «x кратно 3», $Q(x)$ = «x больше 4».

11 $\neg P(x) \wedge \neg Q(x)$, где $P(x)$ = «x не превышает 6», $Q(x)$ = «x не кратно числу 3».

12 $P(x) \leftrightarrow \neg Q(x)$, где $P(x)$ = «x больше 6», $Q(x)$ = «x делится на 3».

13 $\neg P(x) \wedge Q(x)$, где $P(x)$ = «x не достигает 9», $Q(x)$ = «x не превышает 7».

14 $\neg P(x) \oplus Q(x)$, где $P(x)$ = «x – делитель числа 144», $Q(x)$ = «x не больше 6».

15 $P(x) \to \neg Q(x)$, где $P(x)$ = «x не меньше 6», $Q(x)$ = «x четное».

16 $\neg P(x) \leftrightarrow Q(x)$, где $P(x)$ = «x больше 6», $Q(x)$ = «x делится на 3».

17 $\neg P(x) \wedge Q(x)$, где $P(x)$ = «x =< 9», $Q(x)$ = «12 кратно числу x».

18 $P(x) \oplus \neg Q(x)$, где $P(x)$ = «x – делитель числа 125», $Q(x)$ = «x не больше 6».

19 $\neg P(x) \leftrightarrow \neg Q(x)$, где $P(x)$ = «число 48 кратно числу x», $Q(x)$ = «$x - 5 > 0$».

20 $P(x) \wedge Q(x)$, где $P(x)$ = «x + 100 =< 105», $Q(x)$ = «x превышает число 6».

Пример

$P(x) \sim Q(x)$ $M = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

$P(x)$ = «x не больше 6», $Q(x)$ = «x нечетное».

Решение

Предикат принимает значение ИСТИНА на тех элементах множества M, для которых предикаты $P(x)$ и $Q(x)$ либо оба истинны (такими элементами являются 1, 3, 5 – нечетные числа, не превышающие 6), либо оба ложны (выполняется для элемента 8 – четного числа, большего 6).

Таким образом, область истинности данного предиката $I_{P(x)\sim Q(x)} = \{1, 3, 5, 8\}$.

Задание 3

Пусть $x \in$ {люди}, $y \in$ {документы}. На этих областях определения заданы следующие предикаты.

P(x): x – профессор.

V(x): x – поэт.

S(x): x – студент.

K(y): y – конспект.

U(y): y – учебник.

H(y): y – шпаргалка.

N(y): y – роман.

C(y): y – стихи.

L(y): y – письмо.

R(x, y): x любит читать y.

W(x, y): x пишет y.

Запишите предложения в виде формул логики предикатов.

1 Все конспекты – учебники.
2 Некоторые романы написаны в стихах.
3 Ни один учебник не написан в стихах.
4 «Евгений Онегин» – это роман в стихах.
5 Те, кто пишет стихи, – поэты.
6 Все студенты любят читать учебники.
7 Некоторые стихи – письма.
8 Никто не любит писать письма.
9 Студент Громов любит читать учебники.
10 Некоторые студенты пишут только шпаргалки.
11 Все поэты любят читать стихи.
12 Некоторые люди пишут стихи.
13 Никто из студентов не пишет учебники.
14 Профессор Державин – поэт.
15 Поэты пишут только стихи.
16 Все студенты пишут конспекты.
17 Некоторые профессора, а также студенты пишут стихи.
18 Ни один профессор не пишет шпаргалки.
19 Есенин писал только стихи.
20 Каждый, кто любит читать какие-либо стихи, любит читать Пушкина.
21 Все пишут письма.
22 Некоторые не любят читать никаких учебников.
23 Только студенты пишут конспекты.
24 Никто из студентов не пишет никаких учебников.
25 Студенты, которые пишут конспекты, не пишут шпаргалки.

26 Каждый любит читать какие-либо стихи.
27 Некоторые поэты пишут и стихи, и романы.
28 Только стихи пишут только поэты.
29 Поэты пишут стихи.
30 Любой поэт любит читать свои стихи.
31 Профессора пишут учебники.
32 Ни одно письмо не может являться учебником.

Примеры

1 Некоторые учебники представляют собой конспекты $\exists y(U(y)\ \&\ K(y))$.
2 Ни один роман не является учебником $\forall y(N(y) \rightarrow \neg U(y))$.
3 Каждый читает какие-нибудь учебники $\forall x \exists y(U(y)\ \&\ R(x, y))$.
4 Некоторые студенты читают только учебники $\exists x(S(x)\ \&\ \forall y\ (R(x, y) \rightarrow U(y)))$.
5 Пушкин писал и стихи, и романы $\exists y(W(\text{Пушкин}, y)\ \&\ (C(y) \vee N(y)))$.
6 Студент Орлов не пишет письма $S(\text{Орлов})\ \&\ \forall y(L(y) \rightarrow \neg W(\text{Орлов}, y))$.
7 Студент Лосев читает только конспекты $S(\text{Лосев})\ \&\ \forall y(R(\text{Лосев}, y) \rightarrow K(y))$.
8 Любой поэт пишет письма $\forall x(V(x) \rightarrow \exists y(L(y)\ \&\ W(x,y)))$.

Контрольные вопросы

1 Что такое предикат, матрица предиката, область истинности предиката?
2 Что такое логика предикатов и как формулируется вопрос о ее разрешимости?
3 Какой смысл имеют кванторы $\forall$ и $\exists$? Запишите определения символически.
4 Приведите основные равносильности логики предикатов.

5 Лабораторная работа № 5. Машина Тьюринга и рекурсивные функции

Цель работы: сформировать умения применять программу машины Тьюринга к входному слову на ленте; научиться строить машину Тьюринга, получать примитивно рекурсивную функцию.

Теоретические сведения

Машина Тьюринга и рекурсивные функции являются примерами формального алгоритма.

Машина Тьюринга состоит из внешней памяти (бесконечной ленты) с заданным внешним алфавитом A и управляющего устройства с набором его состояний (внутренним алфавитом Q).

Работа машины Тьюринга осуществляется посредством команд вида $q_i a_m \rightarrow q_j L a_n$: если управляющее устройство в состоянии q_i обозревает ячейку с символом a_m, то оно меняет свое состояние на q_j, а содержимое ячейки на a_n и сдвигается влево.

Обозначение сдвига вправо – R, не смещаться – S.

Состояние Стоп обозначается как q_0. Машина Тьюринга остановится, когда управляющее устройство придет в состояние Стоп.

Команды машины Тьюринга выполняются в соответствии заданной программой в зависимости от условий: q_i и a_m. Последовательность выполняемых команд вовсе не должна соответствовать их последовательности в программе машины Тьюринга.

Эквивалентной машине Тьюринга формальной моделью алгоритма являются рекурсивные функции Черча.

Теория рекурсивных функций исследует класс вычислимых функций на базе некоторой системы аксиом.

Говорят, что (n+1)-местная функция f получена из n-местной функции g и (n+2)-местной функции h с помощью оператора примитивной рекурсии, если для любых $x_1, \ldots, x_n$, y справедливы равенства

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n);$$
$$\ldots$$
$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)).$$

Причем независимо от числа аргументов в φ рекурсия ведется только по одной переменной y.

Остальные n переменных $x_1, \ldots, x_n$ на момент применения схемы примитивной рекурсии зафиксированы и играют роль параметров.

Пример 1 **–** Машина Тьюринга задается следующей программой:

$q_1 0 \to q_1 0R$

$q_1 1 \to q_2 0R$

$q_2 0 \to q_0 1S$

$q_2 1 \to q_1 0R$

Найти результат применения заданной машины Тьюринга к записи на ленте: 0111010.

Решение

$$0\,\underset{q_1}{\underline{1}}11010 \to 00\,\underset{q_2}{\underline{1}}1010 \to 000\,\underset{q_1}{\underline{1}}010 \to 0000\,\underset{q_2}{\underline{0}}10 \to 0000\,\underset{q_0}{\underline{1}}10.$$

Пример 2 **–** Построить машину Тьюринга, правильно вычисляющую функцию $f(x, y) = x + y$.

Решение

Число n будем записывать с помощью последовательности из n + 1 единицы, поэтому пара (x; y) в начальном состоянии на ленте будет задаваться следующим образом:

$$\underbrace{1 \quad \dots \quad 1}_{x+1\ \text{единица}} \qquad 0 \qquad \underbrace{1 \quad \dots \quad 1}_{y+1\ \text{единица}}$$

q_1

После работы машины Тьюринга на ленте должно быть выходное слово следующего вида:

$$\underbrace{1 \quad \dots \quad 1}_{x+y+1\ \text{единица}} \qquad 0$$

Нужно, чтобы машина Тьюринга вначале, двигаясь вправо, заменяла нуль, разделяющий массивы единиц, единицей, затем, пройдя массив единиц, нужно начинать движение влево, заменяя две последние единицы нулями.

Программа такой машины может иметь следующий набор команд:

$q_1 1 \to q_1 1R$,
$q_1 0 \to q_2 1R$,
$q_2 1 \to q_2 1R$,
$q_2 0 \to q_3 0L$,
$q_3 1 \to q_4 0L$,
$q_4 1 \to q_5 0L$,
$q_5 1 \to q_5 1L$,
$q_5 0 \to q_0 0R$.

Пример 3 **–** Найти функцию f, получаемую с помощью операции примитивной рекурсии из функций $g(x) = x$, $h(x, y, z) = zy + z$.

Решение

Функцию f в общем случае строят по схеме:

$$f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n);$$
$$\dots$$
$$f(x_1, \dots, x_n, y+1) = h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)).$$

Функцию f от двух переменных будем строить по схеме:

$$f(x, 0) = g(x),$$
$$\dots$$
$$f(x, y+1) = h(x, y, f(x, y)).$$

$f(x, 0) = x.$
$f(x, 1) = h(x, 0, f(x, 0)) = h(x, 0, x) = x.$
$f(x, 2) = h(x, 1, f(x, 1)) = h(x, 1, x) = 2x.$
$f(x, 3) = h(x, 2, f(x, 2)) = h(x, 2, 2x) = 6x.$
$f(x, 4) = h(x, 3, f(x, 3)) = h(x, 3, 6x) = 24x.$
$\dots$
$f(x, y) = y!x.$

Задание 1

Найдите результат применения машины Тьюринга к записи на ленте K, считая, что управляющее устройство установлено под первой слева единицей и находится в состоянии q_1.

1 а)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 1L$
$q_2 1 \to q_2 1L$
$q_2 0 \to q_0 0R$
K: 0111110

б)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0R$
$q_2 1 \to q_3 1R$
$q_2 0 \to q_2 0R$
$q_3 1 \to q_3 1R$
$q_3 0 \to q_0 0$
K:1100101

2 а)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0R$
$q_2 1 \to q_3 1R$
$q_2 0 \to q_2 0R$
$q_3 1 \to q_3 1\ R$
$q_3 0 \to q_0 0$
K:11001101

б)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 1L$
$q_2 1 \to q_2 1L$
$q_2 0 \to q_0 0R$
K:0110101

3 а)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 1L$
$q_2 1 \to q_2 1L$
$q_2 0 \to q_0 0R$
K:010000110

б)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0R$
$q_2 1 \to q_3 1R$
$q_2 0 \to q_2 0R$
$q_3 1 \to q_3 1R$
$q_3 0 \to q_0 0L$
K:100001

4 а)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 1L$
$q_2 1 \to q_2 1L$
$q_2 0 \to q_0 0R$
K:0111010

б)

$q_1 1 \to q_2 1R$
$q_1 0 \to q_1 0R$
$q_2 1 \to q_1 0R$
$q_2 0 \to q_3 0L$
$q_3 1 \to q_0 1L$
$q_3 0 \to q_3 1$
K:10111001

5 а)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0L$
$q_2 1 \to q_2 1L$
$q_2 0 \to q_0 0$
K:1100110

б)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0R$
$q_2 1 \to q_3 1R$
$q_2 0 \to q_2 0R$
$q_3 1 \to q_3 1\ R$
$q_3 0 \to q_0 0$
K:1100101

6 а)

$q_1 1 \to q_2 1$
$q_1 0 \to q_2 0R$
$q_2 1 \to q_0 1R$
$q_2 0 \to q_1 1L$
K:0111010

б)

$q_1 1 \to q_2 1R$
$q_1 0 \to q_1 0R$
$q_2 1 \to q_1 0R$
$q_2 0 \to q_3 0L$
$q_3 1 \to q_2 1L$
$q_3 0 \to q_0 0R$
K:11110101

7 а)

$q_1 1 \to q_2 1\ L$
$q_1 0 \to q_0 0R$
$q_2 1 \to q_2 1R$
$q_2 0 \to q_1 1R$
K:01101110

б)

$q_1 1 \to q_2 1R$
$q_1 0 \to q_1 0R$
$q_2 1 \to q_1 0R$
$q_2 0 \to q_3 0L$
$q_3 1 \to q_2 1L$
$q_3 0 \to q_0 0$
K:10100101

8 а)

$q_1 1 \to q_2 1\ R$
$q_1 0 \to q_2 0L$
$q_2 1 \to q_2 1R$
$q_2 0 \to q_1 1L$
K:010111010

б)

$q_1 1 \to q_2 1R$
$q_1 0 \to q_1 0R$
$q_2 1 \to q_1 0R$
$q_2 0 \to q_0 0L$
$q_3 1 \to q_2 1L$
$q_3 0 \to q_2 0$
K:11010001

9 а)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0L$
$q_2 1 \to q_0 1L$
$q_2 0 \to q_2 0R$
K:1101001

б)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0R$
$q_2 1 \to q_3 1R$
$q_2 0 \to q_2 0$
$q_3 1 \to q_1 1R$
$q_3 0 \to q_0 0$
K:01111010

10 а)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0R$
$q_2 1 \to q_3 1R$
$q_2 0 \to q_0 0$
$q_3 1 \to q_3 1\ R$
$q_3 0 \to q_0 0$
K:010101110

б)

$q_1 1 \to q_1 1R$
$q_1 0 \to q_2 0L$
$q_2 1 \to q_2 1L$
$q_2 0 \to q_0 0$
K:10011101

Задание 2

Постройте машину Тьюринга, правильно вычисляющую функцию.

1 $f(x) = x + 2$.
2 $f(x) = x - 1$.
3 $f(x) = 10 - x$.
4 $f(x) = x - 2$.
5 $f(x) = 1 + x$.
6 $f(x) = y - x$.
7 $f(x) = x + 3$.
8 $f(x) = y - x + 1$.
9 $f(x, y) = x - y$.
10 $f(x, y) = x - y + 1$.
11 $f(x, y) = x + y - 1$.
12 $f(x) = x + y + 1$.

Задание 3

Найдите функцию f(x, y), получаемую из g(x) и h(x, y, z) с помощью операции примитивной рекурсии.

1 g(x) = x, h(x, y, z) = x + 2z.	7 g(x) = 5, h(x, y, z) = xyz.
2 g(x) = 3x, h(x, y, z) = x + z.	8 g(x) = x + 1, h(x, y, z) = x + 2yz.
3 g(x) = 3 + 2x, h(x, y, z) = 2xy + z.	9 g(x) = 2x, h(x, y, z) = xy + 2z.
4 g(x) = x - 1, h(x, y, z) = xyz + z.	10 g(x) = x + 4, h(x, y, z) = x + y + z.
5 g(x) = x + 2, h(x, y, z) = x + xy + 2z.	11 g(x) = x + 3, h(x, y, z) = x + yz.
6 g(x) = 4x, h(x, y, z) = x + y.	12 g(x) = 2, h(x, y, z) = xy + xz.

Контрольные вопросы

1 Дайте определение формального алгоритма, разработанное А. Тьюрингом.

2 Дайте определение формального алгоритма, принадлежащее А. Черчу.

3 Как связаны эти определения?

4 Что такое рекурсивная функция? Какие есть виды рекурсивных функций?

5 Как они строятся? Какие операторы для этого используются?

6 Какие функции берутся в качестве исходных при построении класса рекурсивных функций?

7 Докажите, что функция следования вычислима по Тьюрингу.

8 Дайте определение оператору суперпозиции.

9 Как по-другому называется этот оператор? Приведите пример сложной функции из алгебры.

10 Приведите тезис Черча.

6 Лабораторная работа № 6. Исследование функции трудоемкости алгоритмов

Цель работы: научиться строить точную функцию трудоемкости нерекурсивного алгоритма, выполнять ее асимптотическую оценку.

Теоретические сведения

Трудоемкость конструкции «Следование» есть сумма трудоемкостей блоков, следующих друг за другом. Таким образом, для вычисления теоретически трудоемкости f_A алгоритма A суммируют трудоемкости последовательно идущих друг за другом блоков кода.

С учетом договоренности о том, какие операции в алгоритме мы считаем элементарными, трудоемкость строки кода

```
a[1] = a[0] + 2*(1 + itog)
```

будет равна 6.

Основными блоками алгоритма являются «Ветвление» и «Цикл».

Теоретическая трудоемкость конструкции If вычисляется как

$$F_{If} = f_{условия} + f_{+} \cdot p_{+} + f_{-} \cdot p_{-} ,$$

где f_{+}, f_{-} – трудоемкости ветвей;

p_{+}, p_{-} – вероятности попадания на конкретную ветвь блока условия.

Проверка условия выполняется в любом случае.

Экспериментальная трудоемкость блока, алгоритма это количество элементарных операций, подсчитанных счетчиком в коде.

Для конструкции If теоретически подсчитанное и полученное экспериментально значение трудоемкости могут отличаться. Дело в том, что теоретически рассчитанная вероятность и вероятность, полученная экспериментально, различаются методом вычисления.

Трудоемкость конструкции For зависит от количества итераций и в общем случае в C# определяется по формуле

$$F_{For} = 2 + 2N + N \cdot f_{тела\ цикла},$$

где N – количество итераций цикла.

Задание

Определите точную трудоемкость алгоритма, анализируя псевдокод, оцените ее асимптотически. Экспериментально в программной реализации выведите результат и значение счетчика трудоемкости.

1 Найти сумму минимальных элементов двух одномерных массивов.

2 Найти максимальный элемент над главной диагональю матрицы.

3 Найти минимальный элемент в 3-м столбце матрицы A(n × 2n).

4 Найти максимальный элемент во 2-й строке матрицы A(nx(n + 2)).

5 Найти максимальный элемент во 2-й и 3-й строках матрицы A(nx(n + 3)).

6 Найти разность максимального и минимального элементов в квадратной матрице.

7 Суммировать матрицы A(n × 2n) и B(n × 2n).

8 Умножить матрицы A(n × n) и B(n × n).

9 Умножить матрицы A(n × 2n) и B(2n × n).

10 Умножить матрицы A(2n × n) и B(n × 2n).

11 Найти матрицу A + 2B для матриц A(n × 4n) и B(n × 4n).

12 Найти матрицу 2A + 3B для матриц A(n × 2n) и B(n × 2n).

13 Найти матрицу 5A + B для матриц A(3n × n) и B(3n × n).

14 Транспонировать матрицу.

15 Транспонировать матрицу A(nx(2n + 1)).

16 Для матрицы A(n × m) посчитать количество ненулевых элементов.

17 В матрице C(m × 4m) вычислить сумму неотрицательных элементов.

18 В матрице A((n + 1) × n) посчитать количество нулевых элементов.

19 Отсортируйте одномерный массив по возрастанию.

20 Отсортируйте квадратную матрицу по возрастанию элементов построчно.

Контрольные вопросы

1 Классифицируйте алгоритмы в зависимости от поведения функции трудоемкости на различных входных данных.

2 Опишите правила нахождения точной функции трудоемкости по псевдокоду алгоритма.

3 Опишите, как найти трудоемкость цикла в C# с учетом использования операции i++.

4 Дайте определение асимптотическим оценкам (с графической интерпретацией и символьной записью).

5 Почему для конструкции If теоретически подсчитанное и полученное экспериментально значение трудоемкости могут отличаться?

6 Есть ли смысл подбирать значение p для F_{If}?

7 Как встроить в If счетчик трудоемкости?

8 Как встроить счетчик трудоемкости в For?

7 Лабораторная работа № 7. Сравнительное исследование функций трудоемкости алгоритмов интервальным анализом

Цель работы: сравнить функций трудоемкости количественно-зависимых алгоритмов для определения предпочтительного алгоритма.

Теоретические сведения

Пусть на интервале [a; b] заданы трудоемкости двух алгоритмов f(n) и g(n) и мера $\pi(f(n), g(n))$ расхождения функций с предельным значением допустимого расхождения значений функций φ. Тогда асимптотические оценки можно записать в следующем виде:

$$f(n) = O_{\varphi}(g(n)), \text{ если } \max\ \{\pi(f(n), g(n))\} \le -\varphi; \qquad (1)$$

$$f(n) = \Omega_{\varphi}(g(n)), \text{ если } \min\ \{\pi(f(n), g(n))\} \ge \varphi; \qquad (2)$$

$$f(n) = \Theta_{\varphi}(g(n)), \text{ если } \max\ |\{\pi(f(n), g(n))\}| < \varphi. \qquad (3)$$

Мера расхождения $\pi(f(n), g(n))$ значений функций должна сама быть функцией антисимметричной, т. е. должно выполняться

$$\pi(f(n), g(n)) = -\pi(g(n), f(n))$$

и эквивалентной, т. е. должно выполняться

$$\pi(f(n), g(n)) = 0 \text{ при } f(n) = g(n).$$

Возьмем в качестве $\pi(f(n),\ g(n))$ функцию, которую можно интерпретировать как угловое расхождение значений аргументов:

$$\pi\big(f(n), g(n)\big) = \operatorname{arctg}\frac{f(n)}{g(n)} - \operatorname{arctg}\frac{g(n)}{f(n)}. \qquad (4)$$

Тогда пороговое значение φ может быть интерпретировано как максимальное значение угла расхождения, для которого алгоритмы $f(n)$ и $g(n)$ равноприменимы на интервале $[a;\ b]$.

Если на некотором подынтервале выполняется условие

$$\varphi + \pi(f(n),\ g(n)) < 0, \qquad (5)$$

то $f(n) = O_\varphi(g(n))$ и предпочтительным на данном подынтервале является алгоритм, имеющий функцию трудоемкости $f(n)$.

Если на подынтервале интервала $[a;\ b]$ выполняется условие

$$\varphi - \pi(f(n),\ g(n)) < 0, \qquad (6)$$

то $f(n) = \Omega_\varphi(g(n))$ и предпочтительным на данном подынтервале является алгоритм, имеющий функцию трудоемкости $g(n)$.

Если на подынтервале выполняется условие

$$|\pi(f(n),\ g(n))| - \varphi < 0\ , \qquad (7)$$

то $f(n) = \Theta_\varphi(g(n))$ и оба алгоритма с точностью до порога φ могут быть использованы на этом подынтервале.

Задание

1 Определите соотношение между функциями трудоемкости $f(n)$ и $g(n)$ на интервале $[a;\ b]$ при $\varphi = \pi/k$ с помощью расчетов методом интервального анализа в MS Excel по выданному варианту (таблица 1).

2 Постройте графики данных функций на указанном интервале для подтверждения вывода.

Таблица 1 – Данные для исследования

Вари-ант	f(n)	g(n)	k	[a; b]
1	$2n^2 + 3n \log n + 12n$	$3n^2 + 13n + 10$	32	[20; 80]
2	$4n^3 + 3n^2 + 7n$	$5n^2 \ln n + n + 25$	64	[100; 150]
3	$n^{3/4} + n + 36$	$e^{1,5n\wedge 0,5}$	18	[50; 80]
4	$2,5n^2 + 19n$	$14n \ln n + 22n$	16	[30; 70]
5	$\ln^2 n + 2,5n^2$	$6,5n^{3/2} + 12n + 110$	24	[40; 100]
6	$n^2 + 19n^{1/2}$	$n^{\ln n/2}$	20	[25; 50]
7	$6 \cdot 2^{n-3}$	$n^4 + 3n^3 + 9n^2 + 25n$	36	[120; 150]
8	$\ln^3 n + 4 \ln^2 n$	$1,5n^3 + 8$	48	[20; 90]

Окончание таблицы 1

Вари-ант	f(n)	g(n)	k	[a; b]
9	$n^2 + \log n + n$	$0{,}6n^2 + 3n$	60	[10; 80]
10	$0{,}3n^3 + n^2 + 5n + 4$	$4\ln n + 2n + 5$	64	[80; 140]
11	$n^4 + 6n^2 + 15n$	$e^n + 2n + 5$	18	[10; 40]
12	$0{,}3e^{n - 19}$	$n^3 + 0{,}5n^2 + 2n$	36	[20; 50]
13	$2e^{n - 1}$	$0{,}4n^4 + 1{,}5n$	50	[10; 40]
14	$3 \cdot 2^{n/2}$	$n^3 + 0{,}7n^2 + 2$	36	[120; 150]
15	$\ln n + 5n + 4$	$1/n + n^2$	18	[100; 140]

Исследование функций трудоемкости f(n) и g(n) интервальным анализом в MS Excel представлено на скриншоте (рисунок 1).

	A	B	C	D	E	F	G	H	I
1	f(n)=1,75*n*n		g(n)=18*n*Ln(n)						
2	n	f(n)	g(n)	arctg (f/g)	arctg(g/f)	Pi(f,g)	f=o(g)	f=teta(g)	f=omega(g)
3	20	700	1078,46	0,57572	0,99507	-0,4194	-0,3212	0,3212	0,5175
4	21	771,75	1150,83	0,59072	0,98007	-0,3894	-0,2912	0,2912	0,4875
5	22	847	1224,05	0,60531	0,96548	-0,3602	-0,262	0,262	0,4583
6	23	925,75	1298,1	0,6195	0,95129	-0,3318	-0,2336	0,2336	0,43
7	24	1008	1372,92	0,63331	0,93748	-0,3042	-0,206	0,206	0,4023
8	25	1093,75	1448,49	0,64675	0,92404	-0,2773	-0,1791	0,1791	0,3755
9	26	1183	1524,79	0,65984	0,91096	-0,2511	-0,1529	0,1529	0,3493
10	27	1275,75	1601,78	0,67258	0,89822	-0,2256	-0,1275	0,1275	0,3238
11	28	1372	1679,43	0,68499	0,88581	-0,2008	-0,1026	0,1026	0,299
12	29	1471,75	1757,73	0,69708	0,87372	-0,1766	-0,0785	0,0785	0,2748
13	30	1575	1836,65	0,70886	0,86194	-0,1531	-0,0549	0,0549	0,2513
14	31	1681,75	1916,17	0,72034	0,85046	-0,1301	-0,0319	0,0319	0,2283
15	32	1792	1996,26	0,73153	0,83927	-0,1077	-0,0096	0,0096	0,2059
16	33	1905,75	2076,93	0,74245	0,82835	-0,0859	0,0123	-0,0123	0,1841
17	34	2023	2158,13	0,75309	0,81771	-0,0646	0,0336	-0,0336	0,1628
18	35	2143,75	[illegible]239,87	0,76348	0,80732	-0,0438	0,0543	-0,0543	0,142
19									

Рисунок 1 – Обработка результатов исследования

Контрольные вопросы

1 Что обозначено в методике через φ?

2 φ – это числовая мера или угловая?

3 Что означает π(f(n), g(n))? Как оно отличается своим значением от π = 3,1416?

4 По каким вычислительным результатам вы сделали итоговый вывод в лабораторной работе?

5 Охарактеризуйте содержание столбца F на скриншоте.

6 Охарактеризуйте на скриншоте содержание столбцов G, H, I. Поясните названия этих столбцов.

7 Какие оценки функций используются в лабораторной работе?

8 Дайте определение асимптотическим оценкам функций и дополните их графическими иллюстрациями.

9 Почему в качестве меры расхождения между сравниваемыми функциями выбрана функция $\pi(f(n), g(n))$?

10 По какому критерию интервальный анализ отдает предпочтение функции трудоемкости на конкретном подынтервале?

11 Какой смысл в методике интервального анализа имеет φ?

12 Можно ли решить поставленную задачу только графически?

8 Лабораторная работа № 8. Экспериментальное исследование средней трудоемкости алгоритма для задачи поиска максимума в массиве

Цель работы: экспериментально проверить теоретическую оценку трудоемкости алгоритма поиска максимума.

Теоретические сведения

Алгоритм поиска максимума последовательно перебирает элементы массива, сравнивая текущий элемент массива с текущим значением максимума.

Когда просматривается k-й элемент массива, переприсваивание максимума произойдет, если в подмассиве из первых k элементов максимальным элементом является последний.

Очевидно, что в случае равномерного распределения исходных данных теоретическая вероятность этого события равна 1/k.

Тогда в массиве из n элементов общее количество операций переприсваивания максимума (теоретическое значение) определяется по формуле

$$\sum_{i=1}^{n} 1/i = H_n \approx \mathrm{Ln}(n) + \gamma, \tag{8}$$

где H_n – n-е гармоническое число;

$\gamma = 0{,}57$.

Экспериментальные данные о количестве операций переприсваивания можно получить из k испытаний при фиксированном значении количества экспериментов N_e в испытании.

Случайными величинами в эксперименте являются средние значения количества операций присваивания в каждом из k испытаний, для них нужно определить дисперсию и стандартное отклонение.

При повторении серии испытаний для различных значений N_e была обнаружена следующая зависимость - стандартное отклонение средних σ убывает с ростом N_e обратно пропорционально квадратному корню из количества испытаний:

$$\sigma = b / N_e^{0,5}. \tag{9}$$

То есть $N_e^{0,5}$ и σ находятся в отношении обратной пропорциональности.

Это основная зависимость между этими величинами в экспериментах.

Задание

1 Составьте программу поиска максимума в одномерном массиве из 10 случайных чисел, определяющую экспериментально среднее количество операций присваивания в алгоритме, организовав 10 испытаний для количества экспериментов N_e.

Предусмотрите то, что количество экспериментов N_e последовательно принимает ряд значений: $N_e \in \{1\ 000;\ 10\ 000;\ 100\ 000;\ 1\ 000\ 000;\ 10\ 000\ 000\}$.

2 На основании проведенных экспериментов для каждого значения N_e в MS Excel (рисунок 2) вычислите дисперсию и стандартное отклонение средних.

3 По вычисленным стандартным отклонениям средних постройте линейную регрессию и определите значения k и b уравнения регрессии

$$\mathrm{Ln}\,(\sigma) = k \cdot \mathrm{Ln}(Ne) + b, \tag{10}$$

где значение k на основании зависимости (9) должно быть порядка –0,5.

4 На основе уравнения регрессии получите прогноз σ для значения N_e = 10 000 000 и определите ошибку прогноза в процентах.

Рассмотрим программу нахождения среднего количества операций присваивания в алгоритме поиска максимума.

D14 | f_x =СУММ(D4:D13)/10

Книга1 * ×

	A	B	C	D	E	F	G	H
1	Теоретическое значение			2,928968				
2	**Ne**	**1 000**						
3	k	Sigma		D				
4	1	0,06696825		0,00448475				
5	2	0,02603175		0,00067765				
6	3	0,03996875		0,0015975				
7	4	0,06296875		0,00396506				
8	5	0,05803175		0,00336768				
9	6	0,05903175		0,00348475				
10	7	0,05679825		0,00322604				
11	8	0,00796875		0,0000635				
12	9	0,00203175		0,00000413		D	0,00254908	
13	10	0,06796875		0,00461975		Sigma	0,0504884	
14				**0,00254908**				
15	Ne	Sigma						
16	1,00E+03	0,05048843						
17	1,00E+04	0,01435727						
18	1,00E+05	0,00372455						
19	1,00E+06	0,00138375						
20	1,00E+07	0,00037789						
21	Прогнозируемые значения для Sigma по полиномиальной функции							
22	Xp	Sigma e	Sigma p				Для контроля прогноза	
23	1,00E+07	0,00037789	0,00037584					
24	Ln(Xp)	Ошибка прогноза		0,543360%			Yp=K*Xp + B	
25	16,118096						-7,886360	
26	Аппроксимация полиномиальной функцией							
27	X=Ln(Ne)		Y=Ln(sigma)		Y=K*X + B		Анализ	
28	6,907755		-2,986011		-3,030296		K	
29	9,21034		-4,243499		-4,244312		-0,527240	
30	11,512925		-5,592809		-5,458328		B	
31	13,815511		-6,582961		-6,672344		0,611752	

Рисунок 2 – Обработка результатов экспериментов

Листинг

```
Program max;
  Uses crt;
  Const
     np = 10;
  Type
     fix = Longint;
     fp  = Extended;
     Vect= Array [1..100] of fp;
  Var
     n,ne,k          : fix;
     r               : Vect;
     fe,ft,delta     : fp;
     Function Hn(n:fix):fp;
      Var
```

```
   i:fix;
   s:fp;
  Begin
     s:=0.0;
     for i:=1 to n do
      begin
         s:=s+(1.0/(1.0*i));
      end;
     Hn:=s;
  End; { Hn }
  Function Max(n,ne:fix):fp;
  Var
     i,j   : fix;
     sc,mx : fp;
  Begin
    sc:=0.0;
    randomize;
    for i:=1 to ne do
    begin
       for j:=1 to n do
       begin
          r[j]:=random;
       end;
       sc:= sc + 1.0;     { count max }
       mx:= r[1];
       for j:=2 to n do
        begin
           if mx < r[j]
           then
            begin
               mx := r[j];
               sc:=sc+1.0;
            end;
        end;
    end; { for Ne }
    Max:= sc/(1.0*ne);
 End; {Max}
 Procedure Count;
 Begin
    Clrscr;
    Write('n=');
    Readln(n);
    Write('ne=');
    Readln(ne);
    For k:=1 to np do
     begin
```

```
            ft := Hn(n);
            fe := Max(n,ne);
            delta := abs(ft - fe);
            Writeln;
            Writeln('k=',k:12);
            Writeln('ft=',ft:10:8);
            Writeln('fe=',fe:10:8);
            Writeln('delta=',delta:10:8);
            Readln;
         end;
     End; { Count }
Begin
   Count;
End.
```

Рабочий лист в MS Excel должен содержать обработку данных при всех значениях N_e.

В отчете следует привести скриншоты программы непосредственно с теми данными, которые были обработаны в MS Excel.

Контрольные вопросы

1 Как соотносятся циклы реализации испытаний и экспериментов?

2 К какому классу задач по сложности относится рассмотренная задача?

3 Опишите ситуации худшего, лучшего и среднего случаев для данной задачи.

4 Что такое дисперсия и стандартное отклонение средних?

5 Как в MS Excel построить линейную регрессию?

6 Поясните смысл построения прогноза σ для значения N_e = 10 000 000. Как прогноз связан с линейной регрессией?

7 Что такое аппроксимация?

8 Какая функция называется полиномиальной?

9 Полиномом какой степени аппроксимируется исходная функция?

9 Лабораторная работа № 9. Анализ сложности алгоритмов сортировки

Цель работы: проанализировать сложность рекурсивного алгоритма сортировки вставками.

Теоретические сведения

Время работы рекурсивных алгоритмов складывается из:

1) времени D(n) разбиения исходной задачи на подзадачи (это действие выполняется на входе в рекурсию);

2) $T(n_1) + \ldots + T(n_k)$ – времени решения каждой подзадачи в отдельности (количество подзадач k определяется числом рекурсивных вызовов);

3) S(n) – времени на соединение полученных решений (это действие выполняется на выходе из рекурсии).

Рассмотрим рекурсивный алгоритм сортировки слиянием.

Листинг

```
Type OdMas=Array[1..1000] Of Integer;
Var A : OdMas;
    n : Integer;
…
Procedure Sli(l, s, r : Integer);
Var B : OdMas;
    i, j, k : Integer;
Begin
k:=l; i:=l; j:=s; B:=A;
Repeat
 If A[i]<A[j] Then Begin B[k]:=A[i]; Inc(i) End
              Else Begin B[k]:=A[j]; Inc(j) End;
  Inc(k);
 Until (i>s) Or (j=r);
 If i>s Then
   For i:=j To r Do Begin B[k]:=A[i]; Inc(k) End
  Else For j:=i To s Do Begin B[k]:=A[j]; Inc(k) End;
 A:=B
End;
Procedure Sort_Sli(l, r : Integer);
Var s : Integer;
Begin
 If l<r Then Begin
  s:=(l+r) Div 2;
  Sort_Sli(l, s);
Sort_Sli(s+1, r);
  Sli(l, s, r)
 End
End;
Begin {основная программа}
 …
```

```
 Sort_Sli(1, n);
 …
End.
```

Процедура Sl (процедура слияния отсортированных частей массива) – нерекурсивная процедура, следовательно она анализируется ранее рассмотренным методом. Временная сложность данной процедуры – Θ(n).

Процедура Sort_Sl (основная процедура сортировки слиянием) – рекурсивная процедура. Определим временную сложность данной процедуры.

Очевидно, что если массив состоит из одного элемента (n = 1), то выполняется только проверка условия (l < r), следовательно время работы пропорционально единице T(n) = Θ(1).

Если же в массиве более одного элемента (n > 1), то выполняются следующие действия:

```
     If l<r Then Begin
        s:=(l+r) Div 2;     {1 - Действия  на  входе
в  рекурсию, разбиение  исходной  задачи
на подзадачи}

         Sort_Sli(l, s);
        Sort_Sli(s+1, r);     {2 - два рекурсивных  вызова}

        Sli(l, s, r); {3 - действия  на  выходе
из  рекурсии}
       End
```

Здесь инструкция кода 1 – это разбиение исходной задачи на подзадачи, т. е. D(n) = Θ(1).

Инструкция кода 2 означает, что за два рекурсивных вызова задача разбивается на две подзадачи равной длины T(n/2) + T(n/2) = 2·T(n/2).

Инструкция кода 3 – выполненное на выходе из рекурсии соединение полученных решений (процедура слияния), т. е. S(n) = Θ(n).

Таким образом, время работы данной процедуры определяется соотношением T(n) = D(n) + T(n/2) + T(n/2) + S(n) = 2·T(n/2) + Θ(1) + Θ(n).

Так как сложность Θ(1) меньше сложности Θ(n), то с учетом подбора констант пропорциональности слагаемое Θ(1) можно отбросить (внести под знак Θ(n)). Таким образом, время работы сортировки слиянием определяется рекуррентным соотношением:

$$T(n) = \Theta(1), \text{ если } n = 1;$$

$$T(n) = 2\ T(0{,}5n) + \Theta(1), \text{ если } n > 1.$$

Рассмотрим метод подстановки решения рекуррентных соотношений. Его идея заключается в нахождении (угадывании) функции f(n) такой, что для любого n (n ≥ 1): T(n) ≤ f(n).

Для подбора функции выполняются следующие действия:

1) определяется вид функции $f(n)$ с предположением, что она зависит от некоторых, пока неопределенных, параметров;

2) подбираются значения параметров так, чтобы для всех $n \geq 1$ выполнялось $T(n) \leq f(n)$;

3) доказывается по математической индукции правильность подбора функции и значений параметров.

Так как на каждом этапе рассматриваемая часть массива сокращается в 2 раза, а при слиянии выполняются действия, время которых пропорционально n, т. е. $S(n) = \Theta(n)$, то следует предполагать, что $f(n)$ – логарифмическая функция.

Пусть $f(n) = a \cdot n \cdot \log_2 n$, где a пока неизвестно. Должно выполняться неравенство $T(n) \leq f(n)$ при любом натуральном n. Проверим его при $n = 1$.

$T(n) = \Theta(1)$, т. е. существует константа c_1, такая, что при любом натуральном n справедливо неравенство $T(n) \leq c_1$.

$$f(1) = a \cdot 1 \cdot \log_2 1 = 0.$$

Таким образом, вид функции подобран неверно, т. к. не выполняется неравенство $T(1) \leq f(1)$. Добавим еще один пока неопределенный параметр b следующим образом:

$$f(n) = a \cdot n \cdot \log_2 n + b.$$

Докажем выполнение неравенства при $n = 1$:

$$f(1) = a \cdot 1 \cdot \log_2 1 + b = b \geq T(1) = c_1 \text{ при } b \geq c_1.$$

Докажем методом математической индукции справедливость для любого натурального n.

1 При $n = 1$ справедливость неравенства доказана выше – база индукции.

2 Предположим, что для всех $k < n$ неравенство $T(k) \leq a \cdot k \cdot \log_2 k + b$ выполняется – индуктивное предположение.

3 Докажем справедливость для n.

Так как $n > 1$, то из рекуррентного соотношения следует, что

$$T(n) \leq 2 \cdot T(n/2) + c_2 \cdot n.$$

Поскольку $n/2 < n$, то для $n/2$ выполняется индуктивное предположение, т. е.

$$T(n) \leq 2 \cdot (a \cdot (n/2) \cdot \log_2(n/2) + b) + c_2 \cdot n = a \cdot n \cdot \log_2 n - a \cdot n + c_2 \cdot n + 2 \cdot b \leq a \cdot n \cdot \log_2 n + b,$$

при $a \geq c_2 + b$.

Таким образом, при $b = c_1$, $a = c_1 + c_2$ для любого натурального n выполняется неравенство

$$T(n) \leq (c_1 + c_2) \cdot n \cdot \log_2 n + c_1,$$

т. е. $T(n)$ имеет верхнюю оценку $O(n \cdot \log_2 n)$.

Задание

Сортировку вставкой можно представить в виде рекурсивной последовательности следующим образом: чтобы отсортировать массив A[1, …, n], сначала нужно выполнить сортировку массива A[1, …, n – 1], после чего в этот отсортированный массив помещается элемент A[n]. Запишите рекуррентное уравнение для времени работы описанной рекурсивной версии алгоритма сортировки вставкой и рассчитайте экспериментально трудоемкость данного алгоритма.

Контрольные вопросы

1 Что называют временной сложностью алгоритмов?
2 Что означает запись $T(n) = \Theta(1)$?
3 В чем отличие в оценивании сложности рекурсивных и нерекурсивных алгоритмов?
4 Опишите метод подстановки решения рекуррентных соотношений.
5 В чем состоит метод математической индукции?

10 Лабораторная работа № 10. Исследование временной сложности алгоритмов

Цель работы: экспериментально определить среднее время выполнения обобщенной элементарной операции в языке высокого уровня и получить прогноз времени выполнения программы для больших размерностей множества исходных данных (в зависимости от типов данных).

Теоретические сведения

Имея аналитическую оценку трудоемкости данного алгоритма $f_a(N)$, ее экспериментальное подтверждение и зная среднее время выполнения обобщенной элементарной операции для данного языка и данного процессора t_ср, можно получить оценку времени выполнения программы:

$$t_a(N) = t\_ср * f_a(N).$$

Для получения средневзвешенной оценки t_ср необходимо получить значения $t_a(N)$ для различных значений N, что определяет следующий алгоритм нахождения среднего времени выполнения элементарной операции:

1) измерение времени $t_a(N)$ для различных значений N;
2) экспериментальное или теоретическое получение функции трудоемкости алгоритма $f_a(N)$;
3) определение t_ср = $t_a(N)/f(N)$ для каждого значения N;
4) усреднение полученных значений t_ср по количеству испытаний.

Эта методика дает возможность провести ряд дополнительных исследований и определить зависимость t_ср от типов данных. Полученные значения t_ср для различных типов данных позволяют при сравнительном анализе выявить качественные различия процессоров при обработке данных различного типа, что может являться основой одного из процессорных тестов.

Пересчет полученных значений t_ср в такты процессора дает возможность провести сравнительный анализ относительной скорости выполнения элементарных операций для различных типов данных у процессоров с различной тактовой частотой и различной внутренней архитектурой.

Таким образом, можно обнаружить преимущества той или иной архитектуры по отношению к данной задаче, т. е. решить проблему выбора наиболее рационального процессора по отношению к данной задаче.

Если тестовый алгоритм обладает достаточной устойчивостью по t_ср (значения t_ср, полученные для различных N, имеют малую дисперсию), то можно прогнозировать поведение программы для больших размерностей исходных данных, что позволяет определить границы применимости данного алгоритма на данном процессоре.

Реализуем программно задачу о сумме, определяющую время своего выполнения.

Задача о сумме формулируется как задача нахождения таких элементов S_j исходного массива S из N чисел, что $\sum S_j = V$.

Задача относится к классу NPC, условия существования решения

$$\text{Min}\ \{S[i],\ i = 1,\ N\} \le V \le \text{Sum}.$$

Поскольку исходный массив содержит N чисел, то проверке на равенство V перебором подлежат следующие варианты решений:

1) V содержит одно слагаемое $\Rightarrow C_N^1 = N$ вариантов;
2) V содержит два слагаемых $\Rightarrow C_N^2 = (N\cdot(N-1))/(1\cdot 2)$ вариантов;
3) V содержит три слагаемых $\Rightarrow C_N^3 = N\cdot(N-1)\cdot(N-2)/(1\cdot 2\cdot 3)$ вариантов и т. д. до проверки одного варианта с N слагаемыми.

Поскольку сумма биномиальных коэффициентов для степени N равна $(1+1)^N = \sum C_N^k = 2^N$ и для каждого варианта необходимо выполнить суммирование (с оценкой O(N)) для проверки на V, то оценка сложности алгоритма в худшем случае имеет вид: $F\hat{}_a(N,V) = O(N\cdot 2^N)$.

Определим вспомогательный массив, хранящий текущее сочетание исходных чисел в массиве S, подлежащих проверке на V – массив Cnt[j]. Элемент массива равен 0, если число S[j] не входит в V, и равен 1, если число S[j] входит в V. Решение получено, если $V = \sum S[j]\cdot Cnt[j]$.

Задание

1 Подберите значение N_0, при котором время выполнения кода составляет порядка 0,1 наносекунды на данном процессоре для целого типа данных.

2 Для $N_0 + 100$, … $N_0 + 900$ выполнить замеры времени выполнения, полученные экспериментальные данные занести в расчетную таблицу.

3 Измените тип данных на Long и провести 10 замеров времени для тех же значений $N_0, \ldots, N_0 + 900$.

4 По известной трудоемкости и полученному времени выполнения программы рассчитайте время выполнения обобщенной элементарной операции t_cp в наносекундах и, зная тактовую частоту процессора, t_cp в тактах процессора.

5 На основе полученных данных рассчитайте средневзвешенное значение t_cp по 10 испытаниям и построить прогноз времени выполнения для значений $N \in [N_0 +1000; N_0 + 1500]$.

Листинг – Программа определения времени выполнения процедуры решения задачи о сумме.

```
PROGRAM TaskSum;
Uses Dos,Crt;
Type
    fp      = Extended;
    Value = Word {Extended} {Longint};
    Loop  = Word;
    Vector = Array [0..100] of Value;
Var
   Number,Count    : Vector;
   N               : Loop;
   V               : Value;
   Flag            : Boolean;
   Tbegin,Tend,Tfm :fp;
   Function Time : Fp;
   Var
      t          : fp;
      h,m,s,s100 : Word;
   Begin
      GetTime(h,m,s,s100);
      t:=h*3600.0;
      t:=t+(m*60.0);
      t:=t+(s*1.0);
      t:=t+( (s100*1.0)/100.0 );
      Time:=t;
   End;
Procedure SetNumber;
   Var
       i : Loop;
   Begin
      ClrScr;
      Write('N=');
      Readln(N);
      Write('V=');
      Readln(V);
```

```
        For i:=1 to N do
         begin
            Number[i]:=Random(100)+50;
            Writeln(Number[i]:5);
         end;
     End;
  Procedure FindSum(N: Loop;
  V: Value; Var S,Cnt: Vector;
  Var Flag: Boolean);
     Var
        i,j : Loop;
        Sum : Value;
     Begin
        Flag:=False;
        i:=1;
        Repeat
         begin
         Cnt[i]:=0; Вспомогательный массив коэффициентов при Sj
для проверки суммы
         i:=i+1;
         end;
        Until i>N;
        Cnt[N]:=1;
        Repeat
         begin
            sum:=0;
            i:=1;
            Repeat
             begin
                sum:=sum + Cnt[i]*S[i];
                i:=i+1;
             end;
            Until i>N;
            If sum = V
              then
               begin
                  Flag:=True;
                  Write('OK');
                  Readln;
                  Halt;
                end;
             j:=n;
             While Cnt[j]=1 do
              begin
                 Cnt[j]:=0;
                 j:=j -1;
```

```
            end;
           Cnt[j]:=1;
        end;
    Until Cnt[0] = 1;
  End;

BEGIN
   Randomize;
   SetNumber;
   Tbegin:=Time;
   FindSum(N,V,Number,Count,Flag);
   Tend:= Time;
   Tfm := Tend-Tbegin;
   Writeln('ta(',N:2,')=',Tfm:8:2,' sec.');
   Readln;
END.
```

Обработка результатов экспериментов и построение прогноза результатов выполняются в MS Excel (рисунок 3).

Функция f(n) подбирается исследователем путем замены коэффициентов так, чтобы значения функции примерно соответствовали полученным экспериментально. Общий вид функции, задающий степень ее роста, будет иметь примерно такой же вид.

Вводится тактовая частота процессора, на котором проводился эксперимент, и на основе полученных данных выполняется п. 4 задания.

C17 fx =8*A17*B17+16*B17-3*A17-12

	A	B	C	D	E	F	G	H
1	f(n)= 8*n*2^n + 16*2^n - 3*n - 12				Процессор		2200	МГц
2								
3		Экспериментальные данные						
4	n	2^n	f(n) теория	f(n) эксперимент	TYPE = WORD	Time -s	top - ns	top - takt
5	16	65 536	9 437 124	9 437 138		0,22	23,31	4,66
6	17	131 072	19 922 881	19 922 988		0,44	22,09	4,42
7	18	262 144	41 942 974	41 942 942		0,93	22,17	4,43
8	19	524 288	88 080 315	88 080 349		1,98	22,48	4,5
9	20	1 048 576	184 549 304	184 549 273		4,17	22,60	4,52
10	21	2 097 152	385 875 893	385 875 854		8,62	22,34	4,47
11	22	4 194 304	805 306 290	805 306 226		18,02	22,38	4,48
12						Средние значения	**22,48**	**4,50**
13								
14	Прогнозируемые результаты							
15	n	2^n	f(n) теория	f(n) эксперимент	Прогноз	Прогноз		
16					h:m:s	sec		
17	23	8 388 608	1 677 721 519		0:00:38	37,72		
18	24	16 777 216	3 489 660 844		0:01:18	78,45		
19	25	33 554 432	7 247 757 225		0:02:43	162,93		
20	26	67 108 864	15 032 385 446		0:05:38	337,93		
21	27	134 217 728	31 138 512 803		0:11:40	700		t эксп
22	28	268 435 456	64 424 509 344		0:24:08	1448,3		37,62
23	29	536 870 912	133 143 986 077		0:49:53	2993,1		
24	30	1 073 741 824	274 877 906 842		1:42:59	6179,3		delta %
25	31	2 147 483 648	566 935 682 967		3:32:25	12745		0,25%
26								

Рисунок 3 – Обработка результатов экспериментов и построение прогноза результатов

Контрольные вопросы

1 Как оценивается алгоритм задачи о сумме в худшем случае?

2 Опишите идею реализации полного перебора вариантов достижения суммы.

3 Поясните содержание ячейки A1.

4 Будет ли у вас содержание ячейки A1 таким же?

5 Объясните смысл данных, хранящихся в столбце B.

6 Как заполняются столбцы C и D?

7 Поясните смысл числовых значений в столбцах F, G, H.

8 Как выполнено построение прогноза, вручную или с использованием встроенных средств?

9 Какой смысл имеет величина delta?

10 Какой тип данных в C# соответствует типу WORD в языке Pascal?

11 Какой тип данных в C# соответствует типу LONG в языке Pascal?

12 Сравните результаты, полученные при работе программы с двумя типами данных.

Список литературы

1 **Авдошин, С. М.** Дискретная математика. Формально-логические системы и языки / С. М. Авдошин, А. А. Набебин. – Москва : ДМК Пресс, 2018. – 389 с.

2 **Игошин, В. И.** Сборник задач по математической логике и теории алгоритмов: учебное пособие / В. И. Игошин. – Москва: КУРС ; ИНФРА-М, 2019. – 392 с.

3 **Игошин, В. И.** Математическая логика : учебное пособие / В. И. Игошин. – Москва : ИНФРА-М, 2020. – 399 с.

4 **Гуц, А. К.** Математическая логика и теория алгоритмов / А. К. Гуц. – 3-е изд., испр. – Москва : Либроком, 2016. – 117 с.

5 **Таран, Т. А.** Сборник задач по дискретной математике / Т. А. Таран, Н. А. Мыценко, Е. Л. Темникова. – 2–е изд., перераб. и доп. – Киев: Инрес, 2005. – 64 с.

Приложение А
(рекомендуемое)

Общие требования к отчету

Отчет по лабораторным работам № 1–5 принимается в письменном виде, к лабораторным работам № 6–10 – в печатном виде.

Отчеты по лабораторным работам № 1–5 выполняются в тетради по дисциплине, на титульном листе которой указываются название учреждения образования, название закрепленной за дисциплиной кафедры, название дисциплины; необходимые данные о студенте: ФИО студента, группа; должность, ФИО преподавателя.

Оформление письменного отчета по лабораторным работам № 1–5 должно включать номер и название лабораторной работы, цель работы и постановку задачи, а также ее решение.

Печатный отчет должен содержать стандартные составные части.

1 Титульный лист (рисунок А.1) с указанием следующих реквизитов: название учреждения образования, название закрепленной за дисциплиной кафедры, номер и название лабораторной работы, название дисциплины, вариант, кто выполнил лабораторную работу (ФИО, группа), кто проверяет работу (должность, ФИО), место и дата составления отчета.

2 Цель работы и постановка задачи.

3 Выполненное задание согласно варианту: код программы, реализующей данный алгоритм, с необходимыми комментариями.

4 Скриншоты с входными и выходными данными. Обычно программа тестируется на нескольких вариантах входных данных для проверки ее корректности.

5 Выводы по теме лабораторной работы.

Отчет оформляется шрифтом гарнитуры Times New Roman, кегль 14 пт, межстрочный интервал – полуторный, абзацный отступ – 1,25 см.

Страницы должны быть пронумерованы вверху посередине. Титульный лист при нумерации считается, но не нумеруется.

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ЛОГИКА И
ТЕОРИЯ АЛГОРИТМОВ

Лабораторная работа № 7

Сравнительное исследование функций трудоемкости алгоритмов интервальным анализом

Выполнил: Ковалев В.
ПИР-191

Проверила: Беккер И. А.

Могилев 2022

Рисунок А.1 – Титульный лист отчета