

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Алтайский государственный технический университет
им. И.И. Ползунова»

Бубнова Н.Д.

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

**Методические указания к выполнению лабораторных работ и расчетного задания
для бакалавров по направлению подготовки
09.03.04 «Программная инженерия»**

Барнаул · 2015

Бубнова Н.Д. Алгоритмы и структуры данных [Электронный ресурс]: Методические указания к выполнению лабораторных работ и расчетного задания / Н.Д. Бубнова.- Барнаул : АлтГТУ , 2015 - 30 с. Материалы практикума составлены с учетом требований ФГОС ВО по направлениям подготовки 09.03.04 «Программная инженерия» квалификация (степень) бакалавр. Отражают стандарт учебной дисциплины «Алгоритмы и структуры данных»

Рассмотрены и одобрены
на заседании кафедры
«Прикладная математика»
Протокол № 1от «15» сентября 2015г

ВВЕДЕНИЕ

На лабораторных работах студенты должны выполнять индивидуальные задания на компьютере и сдавать выполненные работы. Сдача работы включает в себя следующие этапы (для конкретной работы могут присутствовать не все этапы):

- показ работоспособности программы на компьютер;
- сдача письменного отчета по лабораторной работе;
- устно-письменная защита и(или) тестирование как по конкретной лабораторной работе, так и по всей теме, которой работа посвящена.

При выдаче индивидуальных заданий к лабораторным работам может учитываться текущий уровень подготовленности студента (задания по одной теме могут отличаться по сложности выполнения).

Номер варианта лабораторной работы, который необходимо выполнить, задается преподавателем в начале занятия.

Лабораторная работа №1

Внутренняя сортировка

Задание

1. Ознакомиться с постановкой задачи
2. Ввести с клавиатуры число исходных записей(число записей не превышает 1000).
3. Написать программу для генерации исходных записей, каждая из которых состоит из пяти полей. Два поля –строковые, три- числовые. Сгенерированные данные поместить в файл input.dat.
4. Ввести с клавиатуры номера тех трех полей, по которым будет выполняться сортировка.
5. Реализовать сортировку предложенным методом. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий.

1. Прямое слияние .
2. Просеивание через пирамиду.
3. Реализовать сортировку прямым включением и сортировку прямым выбором и сравнить время.
4. Шейкерная сортировка.
5. Быстрая сортировка.
6. Сортировка с помощью дерева.
7. Сортировка с помощью разделения.
8. Сортировка с уменьшающимися расстояниями.
9. Сортировка с помощью пирамиды
10. Сортировка методом подсчета.

Лабораторная работа №2

Внешняя сортировка

Задание

1. Ознакомиться с постановкой задачи
2. Ввести с клавиатуры число исходных записей(число записей не ограничено).
3. Написать программу для генерации исходных записей, каждая из которых состоит из пяти полей. Два поля –строковые, три- числовые. Сгенерированные данные поместить в файл input.dat.
4. Ввести с клавиатуры номера тех трех полей, по которым будет выполняться сортировка и число, задающее длину начальных отсортированных серий записей для всех методов, кроме метода

- естественного слияния. В методе естественного слияния число задает максимальную длину отсортированной серии. Длина каждой серии определяется с помощью датчика случайных чисел. Для получения отсортированных серий использовать метод, реализованный в предыдущей работе.
5. Реализовать сортировку предложенным методом. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
 6. Оформить отчет

Варианты заданий

1. Прямое слияние.
2. Многофазная сортировка.
3. Естественное слияние с порогом.
4. Естественное слияние со счетчиком
5. Естественное слияние.
6. Сбалансированное многопутевое слияние.
7. Каскадное слияние.
8. R-путевое слияние .

Лабораторная работа №3

Графы. Реализация обходов.

Задание

1. Ознакомиться с постановкой задачи
2. Выбрать способ представления графа в памяти и написать программу для реализации задачи.
3. Продумать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран в графике. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

1. Найти самый длинный путь, соединяющий две данные вершины графа.
2. Написать программу для определения числа путей между двумя заданными узлами по заданной матрице смежности.
3. Написать программу для проверки графа на полноту. Граф задан матрицей смежности.
4. Найти самый длинный простой цикл в графе.
5. Найти кратчайший путь между двумя заданными вершинами сети.
6. Найти все простые пути между двумя заданными вершинами, если граф задан списковой структурой.
7. Найти все простые циклы графа.
8. Узлы направленного ациклического графа можно перенумеровать так, чтобы получившаяся матрица смежности была бы нижней треугольной. Выполнить проверку графа на ацикличность и указать новую нумерацию узлов для ациклического графа.
9. Проверить, будет ли в графе существовать путь, проходящий через все его вершины. Граф задан матрицей смежности.
10. Указать множество вершин графа, доступных из заданной вершины.
11. Написать программу для нахождения критического пути в сети.
12. Граф задан связными списками. Перечислить вершины, в которые можно попасть из заданной.
13. Взвешенный граф задан списковой структурой. Найти минимальный простой путь между двумя заданными вершинами.
14. Проверить, будет ли граф несвязным.
15. Проверить два графа на изоморфность.
16. Проверить, является ли граф, заданный списком, деревом.
17. В графе подсчитать число путей заданной длины, начинающихся в заданной вершине, и

перечислить их.

18. Проверить, принадлежит ли заданная вершина к числу вершин, не достижимых из исходной вершины. Заданная и исходная вершины задаются..

19. Найти самый короткий простой цикл во взвешенном графе. Граф задан взвешенной матрицей смежности.

20. Между девятью планетами Солнечной системы введено космическое сообщение. Ракеты летают по следующим маршрутам: Земля – Меркурий, Плутон – Венера, Земля – Плутон, Плутон – Меркурий, Меркурий – Венера, Уран – Нептун, Нептун – Сатурн, Сатурн – Юпитер, Юпитер – Марс и Марс – Уран. Можно ли добраться с Земли до Марса?

21. Граф задан матрицей смежности. Проверить его на связность, используя обходы.

22. Семеро друзей разъехались в отпуск и договорились, что каждый пошлет открытку трем остальным. Может ли случиться так, что каждый пошлет открытки тем, кто пошлет открытки ему?

23. Имеется группа островов, соединённых мостами так, что от каждого острова можно добраться до любого другого. Турист обошёл все острова, пройдя по каждому мосту ровно один раз. На острове Троекратном он побывал трижды. Сколько мостов ведёт с Троекратного, если турист

а) не с него начал и не на нём закончил?

б) с него начал, но не на нём закончил?

в) с него начал и на нём закончил?

24. Художник-авангардист нарисовал картину "Контур квадрата и его диагонали".

Мог ли он нарисовать свою картину не отрывая карандаша от бумаги и не проводя одну линию дважды?

25. Дан кусок проволоки длиной 120 см. Можно ли, не ломая проволоки, изготовить каркас куба с ребром 10 см?

26. Пешеход обошёл шесть улиц одного города, пройдя каждую ровно два раза, но не смог обойти их, пройдя каждую лишь раз. Могло ли это быть?

27. Имеется лабиринт состоящий из комнат и переходов. Вам, как организатору развлекательного шоу, необходимо расположить в стартовой комнате N монет, в соседних с ней N-1, в комнатах, в которые можно попасть из этих комнат (исключая те, в которых уже лежат монеты) N-2 и т.д. до 1 (меньше 1 монеты класть нельзя). Естественно, требуется посчитать общую сумму требуемых денег. Лабиринт задан в виде списка коридоров.

Лабораторная работа №4

Графы. Построение каркасов.

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Предусмотреть графическую иллюстрацию результатов.
7. Оформить отчет

Варианты заданий

1. В Алтайский государственный Технический Университет приезжает президент России, он хочет осмотреть все помещения университета. Там, где он пойдет должна быть постелена дорожка. Вам требуется определить по какому маршруту следует расстелить дорожки так, чтобы их суммарная длина была минимальна (деньги). Входные данные заданы в «input.txt». На первой строке содержится количество комнат. Далее каждая строка содержит описание одной комнаты. Сначала следуют координаты комнаты, далее следует количество коридоров выходящих из этой комнаты и следуют номера комнат, с которыми эти коридоры соединят текущую комнату. Нумерация комнат начинается с 0. В выходном файле «output.txt» должны содержаться коридоры, по которым необходимо проложить дорожку (коридоры указываются номерами комнат).

2. В некоторой деревне скучают люди. Чтобы разнообразить свою жизнь они распространяют

слухи. Для каждого жителя известно кому он передает слух, попавший к нему (возможно, нескольким людям). Если слух попадет обратно к тому, кто его «запустил», тогда он обрастает новыми деталями и распространяется дальше. Определить максимальный круг людей, внутри которого циркулирует одна сплетня.

Во входном файле, в каждой строке два числа. Пара чисел a и b в строке означает, что слух от жителя a попадает к жителю b .

3. В далекой стране не очень большой бюджет, но власти хотят, чтобы все города были соединены дорогами. От Вас требуется указать властям между какими городами нужно строить дороги. Входные данные заданы в «input.txt». На первой строке содержится количество городов. Далее следует матрица смежности, в которой 0 означает невозможность построить дорогу, а другое число означает стоимость постройки. В выходном файле «output.txt» должны содержаться дороги, которые необходимо построить (дороги указываются номерами городов). Нумерация городов начинается с 0.

4. Найти базовые циклы в графе, заданном списком дуг

5. Правительство далекой страны не может себе позволить содержать все аэропорты при все повышающихся ценах на бензин. И было принято решение закрыть несколько аэропортов, но не все аэропорты можно закрывать, так как при их удалении клиенты государственной авиакомпании не смогут попасть из любого города в любой. Укажите те аэропорты, которые нельзя закрывать. Список рейсов задан в файле «input.txt».

6. Представьте, что Вы начальник отдела автоматизации новой фирмы, Вам предстоит соединить компьютеры в сеть. Но разумеется начальник не будет Вами доволен, если Вы не сделаете это за минимальную цену. Входные данные заданы в «input.txt». На первой строке содержится количество компьютеров. Далее каждая строка содержит описание одного компьютера. Сначала следуют координаты компьютера, далее следует количество возможных сетевых трасс от этого компьютера и следуют номера компьютеров, с которыми эти трассы соединяют текущий компьютер. Нумерация компьютеров начинается с 1. В выходном файле «output.txt» должны содержаться трассы, по которым необходимо проложить сетевые кабели (трассы указываются номерами компьютеров).

7. Вы нарядили новогоднюю елку, и выяснили, что гирлянды не соединены проводом. От Вас требуется определить, хватит ли у Вас провода, чтобы их соединить, и если хватит, то как это сделать. Входные данные заданы в «input.txt». На первой строке содержится количество гирлянд. На второй строке содержится длина провода, который Вы имеете. Далее каждая строка содержит описание одной гирлянды. Сначала следуют координаты гирлянды, далее следует количество возможных соединений данной гирлянды с другими, и следуют номера гирлянд, с которыми данная может соединяться. Нумерация гирлянд начинается с 0. В выходном файле «output.txt» должен содержаться 0, если Вам не хватит провода, или должны содержаться соединения (соединения указываются номерами гирлянд)

8. Чтобы заработать побольше денег, необходимо соединить некоторым образом между собой определенные факты, но соединение одного факта с разными имеет различную стоимость и Вам необходимо ее минимизировать. Входные данные заданы в «input.txt». На первой строке содержится количество фактов. Далее следует матрица смежности, в которой 0 означает отсутствие возможности соединения фактов, а другое число означает стоимость соединения. В выходном файле «output.txt» должны содержаться соединения, которые указываются номерами фактов. Нумерация фактов начинается с 0.

9. Дан исходный граф $G = (X, V)$. Построить порождённый подграф $G' = (X', V')$, который получается из исходного после удаления указанных вершин и инцидентных им ребер. Найти в G' кратчайший остов.

10. Между населёнными пунктами края проложены дороги. Протяжённость дорог приведена в таблице. Надо отремонтировать дороги так, чтобы из краевого центра можно было бы попасть в любой другой пункт по кратчайшей дороге. Стоимость ремонта дороги пропорциональна ее длине.

11. Имеется лабиринт, состоящий из комнат и переходов. Организатору развлекательного шоу необходимо расположить в стартовой комнате N монет, в соседних с ней $N-1$, в комнатах, в которые можно попасть из этих комнат (исключая те, в которых уже лежат монеты) $N-2$ и т.д. до 1 (меньше 1 монеты класть нельзя). Естественно, требуется посчитать общую сумму требуемых денег. Лабиринт задан в виде списка коридоров.

12. Челнок Фома хочет найти оптимальный маршрут перелета из города, в котором он проживает, в любой другой. Указать те рейсы, которыми следует летать, чтобы потратить меньше денег. Входные данные заданы в «input.txt». В первой строке содержится количество городов. Далее следует матрица

смежности, в которой 0 означает отсутствие рейса, а другое число означает стоимость перелета. В выходном файле «output.txt» должны содержаться рейсы (рейсы указываются номерами городов). Нумерация городов начинается с 1.

13. Имеется лабиринт состоящий из комнат и переходов. Вам, как организатору развлекательного шоу, необходимо расположить в стартовой комнате N монет, в соседних с ней $N-1$, в комнатах, в которые можно попасть из этих комнат (исключая те, в которых уже лежат монеты) $N-2$ и т.д. до 1 (меньше 1 монеты класть нельзя). Подсчитать, сколько монет требуется положить в стартовую комнату для того, чтобы в самых дальних комнатах лежало по 2 монеты (расход монет должен быть минимальным). Лабиринт задан матрицей смежности

14. Найти базовые циклы в графе, заданном списком ребер.

15. Заданы тропинки (координаты вершин и имена вершин, соединенных тропинками). Можно ли проложить еще одну тропинку так, чтобы суммарная длина тропинок была бы минимальной и существовали бы пути до всех выходов. Вход – один. Выходы заданы.

16. Задана карта железнодорожных путей. Встал вопрос о модернизации и ремонте дорог. Как спланировать работу, если проект должен быть минимальным по стоимости и все ранее достижимые станции должны оставаться достижимыми.

17. В городе Маленьком 15 телефонов. Можно ли их соединить проводами так, чтобы затратить минимальное количество проводов.

18. В стране Семерка 15 городов, каждый из городов соединен дорогами не менее, чем с семью другими. Расстояния известны. Какое количество дорог стоит оставить, чтобы из каждого города можно было добраться в любой другой и содержание дорог обходилось бы в минимально возможную сумму.

Лабораторная работа №5

Графы. Независимое множество. Доминирующее множество. Вершинное покрытие.

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Предусмотреть графическую иллюстрацию результатов.
7. Оформить отчет

Варианты заданий

1. Найти минимальные доминирующие множества и числа доминирования для графов правильных многогранников.
2. Шахматному коню можно поставить в соответствие граф, вершины которого расположены на 64 полях доски, а ребра соответствуют ходам этой фигуры. Определить число независимости для графа данной фигуры.
3. Шахматному коню можно поставить в соответствие граф, вершины которого расположены на 64 полях доски, а ребра соответствуют ходам этой фигуры. Определить число доминирования для графа данной фигуры
4. Разместить на шахматной доске минимальное число ферзей так, чтобы они держали под боем каждую клетку доски.
5. Ладейный граф $n \times m$ представляет допустимые ходы ладьи на доске $n \times m$. Вершинам графа можно задать координаты (x, y) , где $1 \leq x \leq n$ и $1 \leq y \leq m$. Какому количеству ребер инцидентна вершина i, j ?
6. Ладейный граф $n \times m$ представляет допустимые ходы ладьи на доске $n \times m$. Вершинам графа можно задать координаты (x, y) , где $1 \leq x \leq n$ и $1 \leq y \leq m$. Какому количеству циклов из четырех вершин принадлежат две несмежные вершины?
7. Определить число доминирования для ладейного графа на доске $m \times n$

8. Выбрать в неориентированном графе минимальное (по количеству вершин) вершинное покрытие.
9. Найти клику в неориентированном графе.
10. Имеется n проектов, которые должны быть выполнены. Для выполнения проекта x_i требуется некоторое подмножество R_i наличных ресурсов из множества $\{1, \dots, p\}$. Пусть каждый проект, задаваемый совокупностью средств, необходимых для его реализации, может быть выполнен за один и тот же промежуток времени. Построим граф G , каждая вершина которого соответствует некоторому проекту, а ребро (x_i, x_j) наличию общих средств обеспечения у проектов x_i и x_j . Какое максимальное множество проектов, которое можно выполнить одновременно за один и тот же промежуток времени?

Лабораторная работа №6

Нахождение максимального потока.

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Предусмотреть графическую иллюстрацию результатов.
7. Оформить отчет

Варианты заданий

1. Найти максимальный поток методом Форда-Фалкерсона. Найти минимальный разрез.
2. Найти максимальный поток методом Форда-Фалкерсона. Найти все разрезы и их пропускную способность
3. Найти максимальный поток методом Форда-Фалкерсона. Указать остаточную сеть.
4. Найти максимальный поток методом Форда-Фалкерсона. Указать дополняющие пути.
5. Найти максимальный поток методом Форда-Фалкерсона. Указать минимальную и максимальную пропускные способности.
6. Найти максимальный поток методом Форда-Фалкерсона. Найти минимальный и максимальный разрезы.
7. Найти максимальный поток методом Эдмондса-Карпа и минимальный разрез.
8. Найти максимальный поток методом Эдмондса-Карпа и длины всех дополняющих путей.
9. Найти методом Форда-Фалкерсона максимальное паросочетание в двудольном графе.
10. Найти максимальный поток методом проталкивания предпотока.
11. Найти максимальный поток методом проталкивания предпотока. Найти число подъемов.
12. Найти максимальный поток методом проталкивания предпотока. Напечатать высоты вершин на каждом шаге.
13. Найти максимальный поток методом проталкивания предпотока и подсчитать число насыщающих проталкиваний.
14. Найти максимальный поток методом проталкивания предпотока и подсчитать число ненасыщающих проталкиваний.
15. Найти максимальный поток методом проталкивания предпотока и подсчитать общее число операций подъема и проталкивания.
16. Найти максимальный поток методом проталкивания предпотока и найти минимальный разрез в сети.
17. Найти максимальный поток методом проталкивания предпотока и расстояние между заданными вершинами в остаточной сети.
18. Найти максимальный поток методом "поднять и в начало".
19. Найти максимальный поток методом "поднять и в начало". Перечислить допустимые ребра.
20. Найти максимальный поток методом "поднять и в начало". Напечатать списки соседей.

21. Найти максимальный поток методом "поднять и в начало". Перечислить все переполнявшиеся вершины.
22. Найти максимальный поток методом "поднять и в начало". Увеличить пропускную способность заданного ребра на 1 и подсчитать максимальный поток. Сравнить число операций.
23. Найти максимальный поток методом "поднять и в начало". Уменьшить пропускную способность заданного ребра на 1 и подсчитать максимальный поток. Сравнить число операций.
24. Найти максимальный поток методом "поднять и в начало". Увеличить пропускную способность заданного ребра в s раз и подсчитать максимальный поток. Сравнить число операций.

Лабораторная работа №7

Раскраска графов.

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Предусмотреть графическую иллюстрацию результатов.
7. Оформить отчет

Варианты заданий

1. Найти хроматическое число графа
2. Найти хроматический индекс графа
3. Найти хроматическое число графа
4. Найти хроматическое число двудольного графа
5. Найти хроматический индекс двудольного графа
6. Найти хроматическое число полного графа
7. Найти хроматический индекс полного графа
8. Найти хроматическое число плоского графа
9. Найти хроматический индекс плоского графа
10. Найти хроматическое число заданного графа, используя алгоритм нахождения независимых множеств, указать, какие вершины в какой цвет окрашиваются.
11. Найти хроматический индекс заданного графа, используя алгоритм нахождения независимых множеств, указать, какие ребра в какой цвет окрашиваются.

Лабораторная работа №8

Потоки минимальной стоимости

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Предусмотреть графическую иллюстрацию результатов.
7. Оформить отчет

Варианты заданий

1. Дана сеть G , состоящая из N вершин и M рёбер. У каждого ребра указана пропускная

способность (целое неотрицательное число) и стоимость единицы потока вдоль этого ребра (некоторое целое число). В графе указан исток S и сток T. Дается некоторая величина K потока, требуется найти поток этой величины, причём среди всех потоков этой величины выбрать поток с наименьшей стоимостью.

2. Найти максимальный поток наименьшей стоимости

3. Вы владелец некоторого завода, выпускающего «Товар», и недавно Вам посчастливилось заключить контракт с одной крупной фирмой, находящейся в другом городе на поставку товаров в их розничную сеть. Так как он находится очень далеко (во Владивостоке), товары придется доставлять авиаперевозкой. В ходе телефонных переговоров Партнер поинтересовался «а на какой объем поставок в день мы можем рассчитывать?». Вы задумались. У Вас есть собственные грузовики (дальнобойщики) осуществляющие транспортировку. Аэропорт находится далеко. Просмотрев накопленную статистику перевозок, Вы выявили, что в собственной области при транспортировке есть некоторые ограничения: на дорогах стоят пункты досмотра груза, весового контроля, некоторые дороги и вовсе ремонтируются. Все это назовем «пропускной способностью» дорог в день. Отталкиваясь от этих условий Вам необходимо узнать: сколько ящиков «Товара» в день вы можете подвозить в аэропорт? При этом, вы хотите эффективно вести бизнес и доставлять товар, кратчайшими маршрутами, т.к. это износ шин, механизмов, в общем амортизационные расходы.

Итого: сколько ящиков Вы сможете транспортировать в аэропорт в день, учитывая пропускную способность дорог, при этом, чтобы общее расстояние маршрутов было минимальным?

4. Реализовать алгоритм Басакера-Гоуэна построения потока минимальной стоимости в случае целочисленных пропускных способностей дуг.

5. Реализовать алгоритм Клейна построения потока минимальной стоимости в случае целочисленных пропускных способностей дуг.

6. Представитель бюро путешествий занимается организацией перелета шести туристов, которых необходимо в один из дней отправить самолетом из разных городов по разным маршрутам. Один турист отправляется из Спрингфилда в Рим, двое — из Рима в Париж, трое — из Парижа в Стамбул. В зависимости от маршрута перелет туристов может быть организован через Спрингфилд, Париж, Рим, Стамбул. При чем на рейс Спрингфилд — Париж осталось 2 свободных места, 2 — на рейс Париж — Рим, 2 — на рейс Рим — Стамбул, 1 — на рейс Спрингфилд — Рим, 4 — на рейс Париж — Стамбул и стоимость билетов разная. Каков наилучший с точки зрения затрат вариант перелета группы между указанными пунктами [4]?

7. Представитель бюро путешествий из предыдущего примера ставит перед собой задачу обеспечить доставку наибольшего количества туристов из Спрингфилда в Стамбул через Париж или Рим в течение первого часа, в течение первых двух часов и т. д. вплоть до первых шести часов.

Лабораторная работа №9

Циклы графов

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Предусмотреть графическую иллюстрацию результатов.
7. Оформить отчет

Варианты заданий

1. Найти самый длинный простой цикл в графе.
2. Найти все простые циклы графа.
3. Найти самый короткий простой цикл во взвешенном графе. Граф задан взвешенной матрицей

смежности.

4. В некоторой деревне скупают люди. Чтобы разнообразить свою жизнь они распространяют слухи. Для каждого жителя известно кому он передает слух, попавший к нему (возможно, нескольким людям). Если слух попадет обратно к тому, кто его «запустил», тогда он обрастает новыми деталями и распространяется дальше. Вам требуется определить максимальный круг людей, внутри которого циркулирует одна сплетня.
5. Найти базовые циклы в графе, заданном списком дуг.
6. Город состоит из перекрестков и соединяющих их дорог. Вам требуется определить: сможет ли таксист подзаработать, провезя вас по кругу несколько раз. Найти все кольцевые дороги. Перекрестки все пронумерованы. Город задается матрицей смежности, где каждая дуга – дорога соединяющая перекрестки
7. Найти базовые циклы в графе, заданном матрицей смежности
8. Найти эйлеров цикл.
9. Каково наименьшее число цепей или циклов необходимо для того, чтобы каждое ребро графа G содержалось точно в одной цепи или цикле?
10. Ребрам графа G приписаны положительные веса. Требуется найти цикл, проходящий через каждое ребро графа G по крайней мере один раз и такой, что для него общий вес (а именно сумма величин $\sum c(a_j)$, где число n_j показывает, сколько раз проходило ребро a_j , а $c(a_j)$ — вес ребра) минимален. Очевидно, что если G содержит эйлеров цикл, то любой такой цикл будет оптимальным, так как каждое ребро проходится только один раз и вес этого цикла равен тогда $\sum c(a_j)$.
11. Рассмотрим проблему сбора домашнего мусора. Допустим, что определенный район города обслуживается единственной машиной. Ребра графа G представляют дороги, а вершины — пересечения дорог. Величина $c(a_j)$ — вес ребра — будет соответствовать длине дороги. Тогда проблема сбора мусора в данном районе сводится к нахождению цикла в графе G , проходящего по каждому ребру G по крайней мере один раз. Требуется найти цикл с наименьшим километражем. С учетом того, как на самом деле курсирует машина, возникает задача: найти Q циклов, которые вместе покроют все ребра графа и при этом их суммарная длина будет минимальной. Машина будет ездить по расписанию и в i -ый день объезжать i -ый цикл.
12. Еще две задачи, когда требуется определить маршрут, проходящий хотя бы один раз по каждой из улиц, возникает при доставке молока или почты. Здесь задача состоит в нахождении маршрута, минимизирующего общий километраж (или время, стоимость и т. д.).
13. Разработать наилучший маршрут проверки электрических, телефонных или железнодорожных линий. Проблема инспектирования распределенных систем (лишь некоторые, из которых названы выше) связана с неперменным требованием проверки всех "компонент".
14. Разработать наилучший маршрут проверки телефонных линий.
15. Разработать наилучший маршрут проверки железнодорожных линий.
16. , Разработать наилучший маршрут для проверки работы автоматических вентиляционных устройств.
17. Разработать наилучший маршрут для уборки помещений и коридоров в большом учреждении.
18. Найти гамильтонов цикл в графе.
19. На планете «Cube», имеющей форму куба, города расположены в вершинах куба и в серединах граней. Любые два города, расположенные на одном ребре, соединены дорогой. Также города, расположенные в серединах граней, соединены с вершинами. Космонавт хочет совершить путешествие по этой планете, пройдя по каждой из её дорог ровно один раз. Сможет ли он это сделать?
20. В некоторой стране из каждого города выходит по три железные дороги. Две компании хотят их все приватизировать. Антимонопольный комитет требует, чтобы из каждого города выходили дороги обеих компаний. Докажите, что компании могут договориться так, чтобы требование антимонопольного комитета было выполнено.

Лабораторная работа №10

Красно-черные деревья

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

1. Найти в дереве путь, содержащий минимальное количество красных вершин, и удалить все вершины, принадлежащие этому пути.
2. Найти в дереве путь, содержащий заданную вершину, и удалить все вершины, принадлежащие пути от корня до этой вершины.
3. Найти в дереве вершину, являющуюся отцом вершины с заданным ключом и удалить ее.
4. В каждом узле дерева содержится информация о студенте: фамилия, номер зачетной книжки и пять экзаменационных оценок. Напечатать фамилии студентов со средним баллом не ниже 4.5. Ключом является номер зачетной книжки.
5. Найти в дереве путь, содержащий максимальное количество красных вершин, и удалить черную вершину, принадлежащие этому пути и расположенную ближе всего к корню.
6. Найти в дереве поддерево, корнем которого будет узел с заданным значением ключа и удалить все вершины, принадлежащие самому левому пути этого поддерева.
7. Найти в дереве запись с заданным значением ключа и подсчитать, сколько узлов имеет значение ключа меньше заданного.
8. В каждом узле дерева содержится информация о студенте: фамилия, номер зачетной книжки и пять экзаменационных оценок. Напечатать фамилии студентов со средним баллом ниже 3.5. Ключом является номер зачетной книжки.
9. В каждом узле дерева содержится информация о студенте: фамилия, номер зачетной книжки и пять экзаменационных оценок. Перенести в новое дерево информацию о студентах имеющих хотя бы одну двойку. Ключом является номер зачетной книжки.
10. Создано два красночерных дерева. В каждом узле дерева содержится информация о студенте: фамилия, номер зачетной книжки и пять экзаменационных оценок. Проверить, есть ли в деревьях узлы, содержащие информацию об одном и том же студенте. Если есть, удалить информацию о них из первого дерева.
11. Создано два красночерных дерева. В каждом узле дерева содержится информация о студенте: фамилия, номер зачетной книжки и пять экзаменационных оценок. Проверить, есть ли в деревьях узлы, содержащие информацию об одном и том же студенте. Если есть, создать на основании этой информации новое дерево, удалив ее из исходных деревьев.
12. Создано два красночерных дерева. Пополнить первое дерево недостающей информацией из второго.
13. Создано два красночерных дерева. Пополнить большее по высоте дерево той записью другого, ключ которой имеет максимальное значение из тех ключей, которые не содержатся в пополняемом дереве.
14. Создано два красночерных дерева. Сформировать из уникальных записей деревьев новое дерево.
15. Создано два красночерных дерева. Пополнить большее по весу дерево той записью другого дерева, ключ которой имеет минимальное значение из тех ключей, которые не содержатся в пополняемом дереве.
16. Создано три красночерных дерева. Напечатать те записи, которые принадлежат одновременно всем деревьям.
17. Создано два красночерных дерева. В каждом узле дерева содержится информация о студенте: фамилия, номер зачетной книжки и пять экзаменационных оценок. Напечатать то дерево, средний балл студентов которого выше.

18. Создано два красночерных дерева. В каждом узле дерева содержится информация о городах: название, численность. Найти информацию о заданном городе и напечатать, сколько жителей в нем проживает.

Лабораторная работа №11

Поиск подстрок

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

1. Пусть все символы в образце различны. Сравнить по быстродействию простейший алгоритм и алгоритм Рабина-Карпа. Предложить модификацию простейшего алгоритма, увеличивающую его быстродействие.
2. Пусть алфавит содержит d символов, и пусть текст и образец - случайные строки длины n и m соответственно. Показать, что математическое ожидание числа сравнений символов, проводимых простейшим алгоритмом при выполнении сравнения строк не превышает $2(n-m+1)$.
3. Пусть в образце наряду с символами алфавита Σ может встречаться символ $*$, означающий любую подстроку (в том числе и пустую). Разработать алгоритм для поиска такого образца в исходном тексте.
4. Пусть алфавит содержит d символов, и пусть текст и образец - случайные строки длины n и m соответственно. Подсчитать среднее число холостых срабатываний алгоритма Рабина-Карпа.
5. Пусть алфавит содержит d символов. Обобщить алгоритм Рабина-Карпа на случай поиска одной из k подстрок.
6. Обобщить алгоритм Рабина-Карпа на случай поиска квадрата размером $m \times m$ в матрице размером $n \times n$.
7. Сравнить по быстродействию алгоритм Кнута-Морриса-Пратта и алгоритм Рабина-Карпа при поиске в тексте образца, все символы которого различны.
8. Сравнить по быстродействию алгоритм Кнута-Морриса-Пратта и алгоритм Рабина-Карпа при поиске в тексте одного из k образцов.
9. Разработать алгоритм для нахождения всех вхождений образца P в текст T по префикс-функции строки PT .
10. Сравнить по быстродействию алгоритм Кнута-Морриса-Пратта и алгоритм Бойера-Мура при поиске в тексте образца, все символы которого различны.
11. Сравнить по быстродействию алгоритм Бойера-Мура и алгоритм Рабина-Карпа при поиске в тексте образца, все символы которого различны.
12. Сравнить по быстродействию алгоритм Кнута-Морриса-Пратта и алгоритм Бойера-Мура при поиске в тексте одного из k образцов.
13. Сравнить по быстродействию алгоритм Бойера-Мура и алгоритм Рабина-Карпа при поиске в тексте одного из k образцов.
14. Модифицировать простейший алгоритм поиска для нахождения в тексте строки y^i -строки, состоящей из i повторений подстроки y .
15. Модифицировать алгоритм Кнута-Морриса-Пратта для нахождения в тексте строки y^i -строки, состоящей из i повторений подстроки y .

16. Модифицировать алгоритм Рабина-Карпа для нахождения в тексте строки y^i -строки, состоящей из i повторений подстроки u .
17. Модифицировать алгоритм Бойера-Мура для нахождения в тексте строки y^i -строки, состоящей из i повторений подстроки u .
18. Пусть алфавит содержит d символов. Обобщить алгоритма Рабина-Карпа на случай поиска одной из 2 подстрок, имеющих общий фрагмент.
19. Пусть алфавит содержит d символов. Обобщить алгоритма Бойера-Мура на случай поиска одной из 2 подстрок, имеющих общий фрагмент.
20. Пусть алфавит содержит d символов. Обобщить алгоритма Кнута-Морриса-Пратта на случай поиска одной из 2 подстрок, имеющих общий фрагмент.
21. Пусть алфавит содержит d символов. Обобщить простейший алгоритм на случай поиска одной из 2 подстрок, имеющих общий фрагмент.
22. Обобщить алгоритм Бойера-Мура на случай поиска одной из 2 подстрок, одна из которых является суффиксом другой.
23. Обобщить алгоритм Рабина-Карпа на случай поиска одной из 2 подстрок, одна из которых является суффиксом другой.
24. Обобщить алгоритм Кнута-Морриса-Пратта на случай поиска одной из 2 подстрок, одна из которых является суффиксом другой.
25. Обобщить алгоритм Бойера-Мура на случай поиска одной из 2 подстрок, одна из которых является префиксом другой.
26. Обобщить алгоритм Рабина-Карпа на случай поиска одной из 2 подстрок, одна из которых является префиксом другой.
27. Обобщить алгоритм Кнута-Морриса-Пратта на случай поиска одной из 2 подстрок, одна из которых является префиксом другой.

Лабораторная работа №12

Суффиксные массивы и суффиксные автоматы

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

1. Пусть требуется в тексте t искать строку s в режиме онлайн (т.е. заранее строку s нужно считать неизвестной). Построить суффиксный массив для текста t и найти подстроку s .
2. На запросы сравнения двух произвольных подстрок (т.е. проверка, что первая подстрока равна/меньше/больше второй) дать ответ (использовать суффиксный массив).
3. Требуется по заданной строке s , произведя некоторый её препроцессинг, научиться за $O(\log |s|)$ отвечать на запросы наибольшего общего префикса (longest common prefix, lcp) для двух произвольных суффиксов с позициями i и j .
4. Требуется по заданной строке s , произведя некоторый её препроцессинг, научиться отвечать на запросы наибольшего общего префикса (longest common prefix, lcp) для двух произвольных суффиксов с позициями i и j .
5. Найти количество различных подстрок в строке.
6. Дана строка S , для которой надо построить суффиксный автомат. Для строки P , требуется за $O(|P|)$ проверить, входит ли она в текст S .

7. Дана строка S , для которой надо построить суффиксный автомат. Для строки P , требуется за $O(|P|)$ вывести позицию её первого вхождения в текст S .
8. Дана строка S , для которой надо построить суффиксный автомат. Для строки P требуется найти за $O(|P|)$ количество вхождений её в текст S .
9. По данной строке S найти за $O(|S|)$ наикратчайшую подстроку, не входящую в S в качестве подстроки
10. Найти наименьший циклический сдвиг строки S за $O(|S|)$.
11. Найти количество различных подстрок за $O(|S|)$.
12. Найти суммарную длину всех различных подстрок различных за $O(|S|)$.
13. Дана строка T , и требуется найти все вхождения в неё строки S за $O(|T|)$ при том, что автомат по-прежнему построен для строки S
14. Даны строки S и T , требуется найти их наидлиннейшую общую подстроку (LCS) за $O(|S|+|T|)$.

Лабораторная работа №13

Суффиксные деревья

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

1. Построить суффиксное дерево для одной строки.
2. Для произвольного набора строк построить бор.
3. Найти включения одной строки в текст
4. Найти наибольшую общую подстроку двух или более строк.
5. Нахождение максимальной повторяющейся подстроки
Для данной строки u , $|u| = n > 0$, найти самую длинную подстроку, встречающуюся в u больше одного раза
6. Нахождение общих элементов двух массивов. Даны два возрастающих массива целых чисел $x[1k]$ и $y[1l]$ Найти количество тех целых t , для которых $t = x[i] = y[j]$ для некоторых i и j (Число действий порядка $k+l$).
7. Построить суффиксное дерево используя алгоритм Укконена
8. Построить суффиксное дерево используя алгоритм МакКрейта
9. Решить задачу о быстром поиске в тексте множества заранее неизвестных слов
10. Решить задачу о количестве различных подстрок в данной строке

Лабораторная работа №14

Динамическое программирование

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

1. В состав производственного объединения входят два предприятия, связанные между собой кооперативными поставками. Вкладывая дополнительные средства в целях развития этих предприятий, можно улучшить технико-экономические показатели деятельности производственного объединения в целом, обеспечив тем самым получение дополнительной прибыли. Величина этой прибыли зависит от того, сколько выделяется средств каждому предприятию и как эти средства используются. Считая, что на развитие i -го предприятия в начале k -го года выделяется a_{ik} ден.ед., найти такой вариант распределения средств между предприятиями в течение N лет, при котором обеспечивается получение за данный период времени максимальной прибыли.
2. Посчитать число последовательностей нулей и единиц длины n , в которых не встречаются две идущие подряд единицы.
3. Дано прямоугольное поле размером $n \times m$ клеток. Можно совершать шаги длиной в одну клетку вправо или вниз. Посчитать, сколькими способами можно попасть из левой верхней клетки в правую нижнюю.
4. Дано прямоугольное поле размером $n \times m$ клеток. Можно совершать шаги длиной в одну клетку вправо, вниз или по диагонали вправо-вниз. В каждой клетке записано некоторое натуральное число. Необходимо попасть из верхней левой клетки в правую нижнюю. Вес маршрута вычисляется как сумма чисел со всех посещенных клеток. Необходимо найти маршрут с минимальным весом.
5. Дана последовательность целых чисел. Необходимо найти ее самую длинную строго возрастающую подпоследовательность.
6. Дана строка из заглавных букв латинского алфавита. Необходимо найти длину наибольшего палиндрома, который можно получить вычеркиванием некоторых букв из данной строки.
7. Из заданного множества предметов со свойствами «стоимость» и «вес», требуется отобрать некое число предметов таким образом, чтобы получить максимальную суммарную стоимость при одновременном соблюдении ограничения на суммарный вес.
8. Имеются монеты достоинством v_1, v_2, \dots, v_n копеек. Необходимо найти наименьшее количество монет, которыми можно выдать сумму S .
9. Имеется судно грузоподъемностью w и n предметов. Известно, что i -ый предмет имеет вес w_i и ценность c_i . Необходимо загрузить судно предметами так, чтобы получить максимальную прибыль.
10. Для умножения матрицы A размера $x \times y$ на матрицу B размера $y \times z$ следует выполнить $x \cdot y \cdot z$ операций. Необходимо вычислить произведение матриц $A_1 \cdot A_2 \cdot \dots \cdot A_n$, сделав при этом минимальное количество операций умножения.
11. По заданным двум символьным последовательностям найти длину наибольшей общей подпоследовательности.
12. X – путешествующий торговец. Он ходит по городам, продавая и покупая товары. Страна, по которой он ходит, состоит из S городов. Начиная свой путь из города S , X должен сделать T переходов между городами. Торговцу разрешается посещать города, в которых он уже был. Свой путь X должен завершить в одном из E городов. Известна прибыль, которую X получает, совершив переход из i -го города в j -ый. Необходимо вычислить максимальную прибыль, которую он может получить, пройдя весь путь.
13. Трамвай должен проехать по маршруту P_0, P_1, \dots, P_n . Расстояние между остановками P_{i-1} и P_i равно S_i . Каждую часть пути S_i водитель должен проезжать с постоянной скоростью v_i , которую он выбирает на остановке P_{i-1} . Обозначим через M_i максимально возможную скорость, которую водитель может выбрать на остановке P_{i-1} ($0 < v_i \leq M_i$). Вероятность поломки трамвая на промежутке S_i равна v_i / M_i . Если поломка имеет место на отрезке S_i , то она случается как раз на середине пути, после чего в течении 10 секунд включается аварийная система и со скоростью 5 метров в секунду трамвай без поломок едет до остановки P_i . Пусть M_0 – максимально допустимая скорость трамвая, с которой он может выехать с начальной станции P_0 . Тогда максимальная скорость, с которой трамвай может выехать со станции P_{i-1} , равна $M_i = M_0 - C_i$, где C_i – суммарное количество поломок на промежутках S_1, \dots, S_{i-1} .
В задаче необходимо вычислить наименьшее среднее время, за которое трамвай может проехать

весь путь от остановки P_0 до P_n .

14. Имеется n чисел. Выберем из них группу из k элементов. Две группы считаются разными, если существует хотя бы один элемент, который принадлежит одной группе и не принадлежит другой. Группирующей системой будем называть множество всех возможных групп. Например, из четырех элементов a, b, c, d можно составить шесть групп по два элемента. Группирующей системой для множества $\{a, b, c, d\}$ и $k = 2$ будет множество $\{ab, ac, ad, bc, bd, cd\}$.

Похожестью группирующей системы называется число, равное сумме произведений всех элементов групп, взятой по модулю m . Похожесть для представленного выше примера для $k = 1$ равна $F_1 = (a + b + c + d) \bmod m$. Для $k = 2$ похожесть равна $F_2 = (ab + ac + ad + bc + bd + cd) \bmod m$.

Для заданного множества чисел и среди всех всевозможных $k = 1, \dots, n$ найти похожесть F_k с максимальным значением.

Лабораторная работа №15

Комбинаторика

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

1. Сколько членов в клубе велосипедистов, если использованы все трехзначные номера, не содержащие 8.
2. Есть участок земли в форме эллипса. На границе участка выбирают n точек. Каждую точку соединяют с остальными прямыми линиями. Какое максимальное число кусков земли можно получить?
3. Пусть X – множество правильно построенных скобочных выражений. Элементы X – строки из открывающихся и закрывающихся скобок. Определение этих строк:
 - пустая строка принадлежит X .
 - если A принадлежит X , то (A) тоже принадлежит X .
 - если A и B принадлежат X , то их конкатенация AB тоже принадлежит X .Длина правильно построенного скобочного выражения E – число скобок в выражении E . Вложенность $D(E)$ множества E определяется правилом:
 - $D(E) = 0$, если E пустое
 - $D(E) = D(A) + 1$, если $E = (A)$ и A принадлежит X .
 - $D(E) = \max(D(A), D(B))$, если $E = AB$ и A и B принадлежат X .Написать программу для расчета длины правильно построенных скобочных выражений и вложенности.
4. Полным k -арным деревом называется такое k -арное дерево, глубина всех листьев которого одинакова и степень ветвления всех внутренних узлов равна k . Для заданных глубины и степени ветвления такого дерева подсчитать число таких способов нумерации узлов, что метка каждого узла меньше меток его потомков. При нумерации дерева с N узлами считать, что разрешены метки от 1 до N .
5. Свете на день рождения подарили k плюшевых игрушки, n мячей и m кукол. Мама положила все игрушки в большую коробку. Сколькими способами Света сможет достать из коробки 1 плюшевую игрушку, 1 мяч и 1 куклу?
6. Мисс Марпл, расследуя убийство, заметила выезжающее от дома мистера Дэвидсона такси. Она запомнила первую цифру "2". В городке номера машин были трехзначные и состояли из цифр 1, 2, 3, 4, 5 и 7. Скольких водителей, в худшем случае, ей придется опросить, чтобы найти настоящего убийцу?

7. В соревнованиях по фигурному катанию принимали участие россияне, итальянцы, украинцы, немцы, китайцы и французы. Сколькими способами могут распределиться места по окончании соревнований?
8. Пете на день рождения подарили k новых дисков с играми, а Вале папа привез n дисков из командировки. Сколькими способами они могут обменять 4 любых диска одного на 4 диска другого?
9. В ювелирную мастерскую привезли n изумрудов, k алмазов и m сапфиров. Ювелиру заказали браслет, в котором 3 изумруда, 5 алмазов и 2 сапфиров. Сколькими способами он может выбрать камни на браслет?
10. В кабинете заведующего ювелирного магазина имеется код, состоящий из двух различных гласных букв русского алфавита, за которой следуют 3 различные цифры. Сколько вариантов придется перебрать мошеннику, чтобы раздобыть драгоценности, которые там хранятся?
11. В 9 классе 15 предметов. Завучу школы нужно составить расписание на субботу, если в этот день 5 уроков. Сколько различных вариантов расписания можно составить, если все уроки различные?
12. Жил был хитрый администратор, который знал сколько всего можно придумать паролей из n маленьких русских букв длины n (без повторов). А Вы знаете? Ввод n осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».
13. Опытный взломщик по имени Brute Force решил посчитать, сколько можно придумать паролей из заданного пароля путем перестановки букв. Первоначальный пароль задается в «input.txt». Вывод осуществлять в файл «output.txt».
14. Пример: из пароля «саш» можно получить 6 паролей (считая исходный), т.е. ответ 6.

Лабораторная работа №16

Вычислительная геометрия

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Предусмотреть графическую иллюстрацию результатов.
7. Оформить отчет

Варианты заданий

1. Ввести число точек и сгенерировать их координаты. Найти точки, ближайšie по L_m расстоянию, где $L_m = (|X_1 - X_2|^m + |Y_1 - Y_2|^m)^{1/m}$.
8. Ввести число точек и сгенерировать их координаты. Найти точки, ближайšie по L расстоянию, где $L = \max(|X_1 - X_2|, |Y_1 - Y_2|)$.
9. Ввести число точек и сгенерировать их координаты. Найти все выпуклые слои.
10. Ввести число точек и сгенерировать их координаты. Найти все максимальные слои. Максимальный слой состоит из мажорирующих точек. Точка (x, y) мажорирует точку (x_1, y_1) , если $x \geq x_1$ и $y \geq y_1$.
11. Заданы два множества точек. Найти расстояние между множествами, если оно находится как расстояние между ближайшими точками разных множеств.
12. Перечислить заданные точки в порядке обхода против часовой стрелки относительно заданного полюса.
13. Определить, лежат ли какие-нибудь 3 из заданных n точек на одной прямой.
14. Определить, лежат ли какие-нибудь 4 из заданных n точек в одной плоскости.
15. Определить, являются ли заданные точки вершинами выпуклого многоугольника. Перечислить их в порядке обхода вершин по или против часовой стрелки, если являются.
16. Задан набор отрезков. Перечислить пересекающиеся.
17. Задан многоугольник. Разбить его на выпуклые многоугольники, если исходный многоугольник

не выпуклый.

18. Задан многоугольник. Проверить, является ли заданная точка внутренней. Исходный многоугольник не обязательно выпуклый.

19. Задан многоугольник. Проверить, является ли заданная точка внешней. Исходный многоугольник не обязательно выпуклый

20. Вычислить площадь заданного многоугольника. Исходный многоугольник не обязательно выпуклый

21. Задана многозвенная ломаная. Проверить, будет ли она самопересекающейся.

22. Заданы две многозвенные ломаные. Проверить, будут ли они пересекающимися.

23. Круги заданы координатами центров и длинами радиусов. Проверить, будут ли среди них пересекающиеся.

24. Среди точек заданного множества найти ближайшие.

25. Среди точек заданного множества найти максимально удаленные.

26. Задана плоскость и множество сфер. Сферы заданы координатами центров и длинами радиусов. Перечислить сферы, пересекающиеся с плоскостью.

27. Задана плоскость и множество сфер. Сферы заданы координатами центров и длинами радиусов. Перечислить сферы, расположенные на максимальном расстоянии от плоскости.

28. Задана две параллельные плоскости и множество сфер. Сферы заданы координатами центров и длинами радиусов. Перечислить сферы, расположенные вне пространства между плоскостями.

Лабораторная работа №17

Б-дерева порядка n

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

Задание

1. На основании данных, содержащихся в файле input.dat , сформировать Б дерево порядка n .
 2. Полученное дерево отобразить на экране.
 3. Реализовать предложенные в варианте действия и результат отобразить на экране.
- Сформировать файл output.dat. Поместить в него исходное дерево и дерево, полученное после предложенных в варианте действий.

Варианты заданий

1. Обойти Б-дерево порядка n сверху вниз.
2. Обойти Б-дерево порядка n слева направо
3. Обойти Б-дерево порядка n справа налево
4. Удалить из Б-дерева информацию по заданному ключу.
5. В Б-дереве найти максимально заполненные узлы.
6. В Б-дереве найти максимальный ключ из ключей, принадлежащих заданному диапазону.
7. В Б-дереве вычислить процент заполненности узлов.
8. В Б-дереве найти минимальный ключ из ключей, принадлежащих заданному диапазону.
9. Проверить, все ли ключи второго дерева содержатся в первом.
10. Удалить из дерева ключи, принадлежащие заданному диапазону значений.
11. Разбить дерево на два по заданному значению ключа (В первое дерево занести записи с ключами меньше заданного, во второе – все остальные).

13. Удалить ключи, принадлежащие диапазону [a,b].
14. Определить процент ключей первого дерева, принадлежащих второму .
15. Удалить из второго дерева ключи, принадлежащие первому.
16. Определить след заданного ключа.
16. Удалить ключ с заданным следом.
17. Проверить, принадлежит ли второе дерево первому.
18. В Б-деревьях содержится информация о файлах и каталогах:
Имя, признак(файл, каталог), размер, дата, ссылка на отца.
Напечатать полное имя файла.
19. Удалить из первого дерева ключи, принадлежащие второму.
20. Для заданного ключа найти предшественника и последователя и исключить их.
21. Подсчитать среднее значение ключей в узле. Предложить, какие ключи надо исключить и в каком порядке, чтобы дерево не изменило высоты и число ключей было бы равно n.
22. Подсчитать число ключей в дереве. Предложить такой порядок поступления ключей, чтобы высота полученного дерева была бы минимальной.
23. Задана очередь, каждый элемент которой содержит вид операции над деревом и необходимую для ее выполнения информацию. Выполнить операции в соответствии с очередью.
24. Удалить ключи так, чтобы дерево стало минимально заполненным без изменения высоты.
25. Задано Б-дерево и некоторое число. Разбить исходное дерево на два так, чтобы часть записей перешла бы в новое дерево, а остальные данные остались бы в исходном. На количество записей в исходном дереве указывает заданное число. Ключи записей нового дерева должны быть меньше ключей записей оставшихся в исходном.
26. Указать, какому диапазону должен принадлежать ключ, добавление которого вызовет минимальную перестройку дерева.
27. Проверить на эквивалентность два Б-дерева. Если они не эквивалентны, указать минимальное по мощности множество ключей которое надо удалить, чтобы деревья стали эквивалентными.
28. Проверить на эквивалентность два Б-дерева. Если они не эквивалентны, указать минимальное по мощности множество ключей которое надо добавить, чтобы деревья стали эквивалентными.
29. Задано Б-дерево. Какие ключи и в каком порядке надо исключить, чтобы число ключей в каждом узле стало критическим и высота дерева уменьшилась бы на единицу.
30. Сколько ключей надо добавить, чтобы высота Б-дерева не изменилась.

Лабораторная работа №18

Длинные числа

Задание

1. Ознакомиться с постановкой задачи
2. Написать программу для ее реализации
3. Разработать тесты
4. Исходные данные поместить в файл input.dat
5. Результаты вывести на экран. Исходные данные и результаты вывести также в выходной файл output.dat
6. Оформить отчет

Варианты заданий

- 1) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

$$\text{где } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \quad B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}.$$

$a_{ij} = i^{67} \cdot j + \frac{S(i)}{e^i}$, $S(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$, $b_i = i + 3$. Ввод n и X осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

2) Вывести в файл результат перемножения матриц АВ, где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}$ $B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

$a_{ij} = \prod_{m=1}^q (i + j + m)^3$, $b_i = i + 333333333333$, q – ввести с клавиатуры. Ввод n осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

3) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}$ $B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

$a_{ij} = \prod_{m=1}^2 \frac{(mi! + mj + m + m)^{1/2}}{\sqrt{m}}$, $b_i = \sum_{m=1}^{10000} mi$. Ввод n и X осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

4) Вывести в файл результат перемножения матриц АВ, где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}$ $B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

$a_{ij} = \prod_{m=1}^{\frac{1}{3} \left(\frac{ij}{i-j} - 2 \left(\frac{j-i}{ij} \right)^{-1} \right)^2 \left(\frac{i^2 j^2}{i^2 - 2ij + j^2} \right)^{-1}} \frac{(i + j + 3)^{1/3}}{0,5}$, $b_i = i^{i^i}$. Ввод n осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

5) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}$ $B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

$a_{ij} = (i + j)^{37}$, $b_i = i$. Ввод n и X осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

6) Вывести в файл результат перемножения матриц АВ, где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}.$

$a_{ij} = (ij)! + e^{\frac{\left(\frac{i+j}{ij}\right)^2 + \left(\frac{i-j}{ij}\right)^2}{i^2+j^2} - \frac{2}{i^2j^2}}, b_i = \frac{i!}{0,125}.$ Ввод n осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

7) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}.$

$a_{ij} = i + j + e^{-\infty}, b_i = \frac{i^{47}}{\sum_{m=1}^{100} 1}.$ Ввод n и X осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

8) Вывести в файл результат перемножения матриц АВ, где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}.$

$a_{ij} = i^j + je^{\sin\left(\frac{\frac{i}{j} + \left(\frac{j}{i}\right)^{-1}}{i} \pi j\right)}, b_i = \frac{i}{\prod_{m=1}^{10} 1}.$ Ввод n осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

9) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}.$

Ввод n, X и коэффициентов a_{ij} и b_i осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

10) Вывести в файл результат перемножения матриц AB , где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}$ $B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

$$a_{ij} = i^{33} + j \sin \left[\pi e^{\frac{\sum_{m=1}^{100000000} m}{\left(\frac{(1+100000000)^{1000000000000}}{2} \right)^{-1}}} - 1 \right], b_i = i!. \text{ Ввод п осуществлять из файла}$$

«input.txt». Вывод осуществлять в файл «output.txt».

11) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

$$\text{где } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \quad B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}.$$

$a_{ij} = tg(2\pi(i+j)) + i \frac{1}{(j!)^{-1}}$, $b_i = \sin(\pi i)$. Ввод n и X осуществлять из файла «input.txt». Вывод осу-

ществлять в файл «output.txt».

12) Вывести в файл результат перемножения матриц AB , где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}$ $B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

$a_{ij} = e^{-tg(\pi(i+j))} + 3ij$, $b_i = 3i^{31}$. Ввод n осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

13) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

$$\text{где } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \quad B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}.$$

$a_{ij} = e^{\ln(\sin(2\pi(i+j)))} + 22ij$, $b_i = i!$. Ввод n и X осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

14) Вывести в файл результат перемножения матриц АВ, где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

n – ввести с клавиатуры, $a_{ij} = e^{\frac{\lg 3}{\lg e}} + i^{32} + j$, $b_i = (i!)!$. Вывод осуществлять в файл «output.txt».

15) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

$a_{ij} = \sin(\arctg(\ln(\cos(2\pi i + \frac{\pi}{2})))) + i^{32} + j$, $b_i = i^3 / i$. Ввод n и X осуществлять из файла «input.txt».

Вывод осуществлять в файл «output.txt».

16) Вывести в файл результат перемножения матриц АВ, где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

n – ввести с клавиатуры, $a_{ij} = e^{\text{ctg}(\arctg(\ln(\cos(2\pi i + \frac{\pi}{2}))))} + i^{31} + j$, $b_i = i + 4$. Вывод осуществлять в файл «output.txt».

17) Проверить является ли данное X решением системы линейных уравнений $AX=B$,

где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

$$a_{ij} = \sin \left(2\pi - \cos(1) \cdot \left(\frac{\sum_{m=1}^{100000} m}{2 + \frac{100000^{10000000000000000}}{i} - (i+1) - \frac{100000^{10000000000000000}}{i} - 1} - 1 \right) \right) + i + j!$$

$$\left(0,5 \frac{\left(100000^{10000000000000000} \right)^{-1}}{i} \right)$$

$b_i = i^3 / i + 4$. Ввод n и X осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

22) Вывести в файл результат перемножения матриц АВ, где $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{pmatrix}$.

n – ввести с клавиатуры, $a_{ij} = \frac{\log_{33} e^{\cos(\pi + \pi(2j+1)) + 33ij}}{\log_{33} e^{\sin(\pi + \pi(2j+2))}} + i + (j^{-1} / i)^{-1}$, $b_i = (i + i + 4)!$. Вывод осуществлять в файл «output.txt».

23) Вычислить количество операций, необходимых для вычисления определителя матрицы n на n разложением по строке. Ввод n осуществлять с клавиатуры. Вывод осуществлять в файл «output.txt».

24) Вычислить количество операций, необходимых для вычисления обратной матрицы n на n. Ввод n осуществлять с клавиатуры. Вывод осуществлять в файл «output.txt».

25) Написать операцию длинного деления с оптимальным выбором начального приближения и без него. Ввод осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

26) Даны всевозможные сочетания длины n из малых букв английского алфавита, отсортированные в лексикографическом порядке. По заданному сочетанию определить его порядковый номер. По заданному номеру вывести соответствующее сочетание. Ввод осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

27) Даны всевозможные перестановки длины n из малых букв английского алфавита, отсортированные в лексикографическом порядке. По заданной перестановке определить ее порядковый номер. По заданному номеру вывести соответствующую перестановку. Ввод осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

28) Даны всевозможные перестановки длины n из малых букв английского алфавита, отсортированные в обратном лексикографическом порядке. По заданной перестановке определить ее порядковый номер. По заданному номеру вывести соответствующую перестановку. Ввод осуществлять из файла «input.txt». Вывод осуществлять в файл «output.txt».

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РАСЧЕТНОГО ЗАДАНИЯ ПО ДИСЦИПЛИНЕ

"АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ"

1 Общие сведения о расчетном задании

1.1 Цель работы по выполнению расчетного задания

Расчетное задание по дисциплине «Алгоритмы и структуры данных» предусмотрено для студентов специальности ПИ в четвертом семестре и является важным этапом в изучении студентами соответствующей учебной дисциплины.

Цель данного расчетного задания:

- систематизация, закрепление и расширение теоретических и практических знаний по дисциплине и применение этих знаний при разработке и создании программного продукта, позволяющего решить задачи, сформулированные в условиях задания;
- развитие навыков выполнения самостоятельной работы, овладение методами исследования и

- экспериментирования при решении вопросов научно-исследовательского характера;
- приобретение навыков по оформлению и представлению результатов проделанной работы.

1.2 Организация проведения расчетного задания

Расчетное задание выполняется согласно заданию, выданному руководителем работы. Задание выдается в начале семестра. В задании указывается тема работы, основные этапы работы и сроки их выполнения, сроки представления работы к защите. Тема предполагает создание программного продукта для решения некоторой конкретной задачи. Информация о темах работ дана ниже в разделе 2.

Студент должен в соответствии с темой разработать программный продукт и оформить пояснительную записку (отчет о проделанной работе). Информация о разработке программного продукта дана ниже в п.1.3. Информация об оформлении отчета дана ниже в разделе 3.

Расчетное задание должно быть выполнено и сдано руководителю до окончания семестра в срок, установленный в задании. По результатам выполнения задания студенту выставляется оценка по пятибалльной шкале (и рейтинг по стобалльной шкале) в зачетной ведомости и в зачетной книжке.

Процесс сдачи расчетного задания включает в себя:

- демонстрацию работы программного продукта на компьютере,
- представление отчета о выполнении работы (пояснительной записки),
- устную защиту.

Студент должен сдать в отдельной папке:

- отчет на бумажном носителе;
- электронную копию отчета;
- программный продукт в электронном варианте.

Устная защита - это выступление студента с докладом по теме работы, возможно с использованием электронной презентации (время доклада – не более 5 минут) и проведение беседы по работе в целом и по тексту программы.

Процесс выполнения задания рекомендуется разделить на следующие этапы:

- 1) выбор темы, развернутая постановка задачи, изучение необходимой учебной и научно-технической литературы (1-4 недели семестра);
- 2) разработка структуры данных и алгоритма решения задачи (3 -6 недели семестра);
- 3) написание текста программы (5-12 недели семестра);
- 4) тестирование и отладка программного продукта (9-12 недели семестра);
- 5) оформление отчета о проделанной работе (13-14 недели семестра);
- 6) сдача работы руководителю и защита работы (14-17 недели семестра).

1.3 Разработка программного продукта

Вопрос о языке и среде программирования решается для каждой работы индивидуально. В некоторых случаях руководитель работы однозначно определяет это в выдаваемом задании, в некоторых случаях студент определяет этот вопрос самостоятельно.

Требования, предъявляемые к программному продукту:

- программа должна быть работоспособной;
- программа должна полностью решать поставленную задачу;
- программа должна правильно решать поставленную задачу;
- программа должна быть защищенной, то есть иметь защиту от неправильного ввода и других возможных нежелательных ситуаций;
- программа должна иметь понятный и удобный интерфейс;
- программа должна быть проверяемой, то есть качества программы возможно продемонстрировать на практике.

При разработке программного продукта следует придерживаться методологии структурного программирования, что означает:

- нисходящее проектирование и кодирование, то есть алгоритм разрабатывается «сверху-вниз», начиная со списка входных и выходных данных с постепенной детализацией алгоритма;
- модульное программирование, то есть разделение программы на отдельные логические части (модули) и последовательное программирование каждой части;

- структурное кодирование, то есть написание хорошо структурированных программ с одним входом и одним выходом на основе ограниченного набора базисных конструкций.

Работу над программным продуктом рекомендуется организовать в следующем порядке: в первую очередь следует создать работающую основу («костяк») программы, не акцентируясь на защитах и интерфейсе, затем следует сосредоточить усилия на защитах, а затем – на интерфейсе. К сожалению, иногда студенты работают в «обратном» порядке, в итоге к окончанию семестра готовы только красивая заставка и главное меню, что не позволит получить даже удовлетворительную оценку.

Следует стремиться к тому, чтобы программный продукт был готов к демонстрации на компьютере примерно за месяц до конца семестра. В этом случае студент сможет несколько раз показать руководителю работу программы и исправить отмеченные ошибки и недостатки. Если программа будет готова только в самом конце семестра, то времени на доработку, которая может потребоваться, не будет, что приведет к снижению итоговой оценки.

2 Темы расчетных заданий

Темы расчетных заданий определяет руководитель работы. Темы должны соответствовать профилю соответствующей учебной дисциплины. Кроме того, при выборе темы следует учитывать индивидуальные пожелания студентов, их предпочтения в плане индивидуальной образовательной траектории и научно-исследовательской деятельности. Так, возможен вариант, когда тема предлагается студентом и после обсуждения с руководителем выдается студенту на проектирование.

Общие требования к теме расчетного задания по дисциплине «Алгоритмы и структуры данных»:

- в рамках реализации предложенной темы необходимо разработать на языке высокого уровня программный продукт для решения определенной задачи;

- реализация предлагаемой темы должна охватывать основные разделы, изученные в данной дисциплине;

- при выполнении задания студент должен продемонстрировать основные знания, умения и навыки, полученные при изучении данной дисциплины

- реализация предлагаемой темы должна предусматривать выполнение студентом некоторого объема поисково-исследовательской работы (самостоятельный поиск информации из литературных источников, в том числе из Интернета, исследование и анализ различных способов решения проблем и т.д.).

Темы могут быть нестандартными (каждая тема требует индивидуального подхода к решению) и стандартными (есть общее для всех студентов задание, а затем для каждого студента указывается некоторая модификация этого задания). Общее задание для стандартных заданий дано ниже в подразделе 2.1..

К постановке задачи по заданной теме студент должен надо относиться творчески, то есть в процессе выполнения задания допустимо вносить, например, изменения в предлагаемое меню, добавлять поля в предлагаемую структуру записи и т.д. Естественно, что эти моменты должны быть согласованы студентом с руководителем работы.

2.1 Задание для стандартных работ

Цель работы: приобрести практический опыт создания программ, использующих файлы, а также приобрести начальные навыки работы с базами данных.

Задание.

Ознакомиться с предметной областью

Сформулировать 8-10 задач, связанных с заданной предметной областью

Обсудить задачи с преподавателем и определить задачи, подлежащие программной реализации.

Определить, какая информация потребуется для решения выбранных задач и как она должна быть организована. Обосновать свой выбор и обсудить результаты с преподавателем.

Разработать программы и представить отчет

Примеры заданий

1. Биржа труда
2. Автоматизированное рабочее место администратора гостиницы
3. Организация соревнований по биатлону
4. Выбор сроков проведения соревнования

Другие варианты заданий:

Разработать программу кодирования-декодирования

Разработать программу для выбора расположения станций оказания экстренной помощи.

Реализовать эффективный алгоритм обращения матриц

Реализовать алгоритм решения систем уравнений для систем высокого порядка

Программа должна быть структурированной. Программа должна быть, по возможности, защищена от ошибок (от ошибок ввода и др.). В программе должен быть реализован достаточно понятный интерфейс. Программа должна быть дружелюбной.

3 Оформление отчета о выполнении расчетного задания

Пояснительная записка расчетного задания (отчет) должна включать в себя следующие структурные единицы:

- титульный лист;
- задание;
- содержание;
- перечень условных обозначений, символов, терминов (при необходимости);
- введение;
- основная часть, разделенная на пронумерованные разделы, подразделы, пункты, подпункты;
- заключение;
- список использованных источников;
- приложения.

Каждая структурная единица отчета начинается с нового листа. В основной части, кроме того, каждый раздел начинается с нового листа.

Содержание отчета («что писать») и правила оформления отчета («как оформлять») подробно представлены учебном пособии.

4 Общие рекомендации для студентов

4.1 На начальном этапе внимательно изучите методические указания к выполнению расчетного задания и соответствующее учебное пособие.

4.2 Ответственно отнеситесь к выбору темы; если у Вас есть какие-то свои пожелания, обязательно сообщите об этом руководителю работы.

4.3 После выбора темы и получения задания составьте свой индивидуальный график работы на основе общих рекомендаций по срокам выполнения отдельных этапов.

4.4 Результаты отдельных видов работ сразу же документируйте. В этом случае к окончанию работы текст пояснительной записки у Вас будет практически готов.

4.5 Детально разберитесь с постановкой задачи (что дано, что должно быть получено, есть ли специальные требования к интерфейсу и т.д.).

4.6 Проведите обзор источников в библиотеке и в Интернете. Оцените суть поставленной задачи, направьте свои усилия на поиск и систематизацию информации о существующей методике решения поставленной задачи или аналогичных задач. В итоге сформулируйте способ решения поставленной проблемы.

4.7 Приступите к разработке программного обеспечения. Более подробно рекомендации по этому этапу работы представлены в подразделе 1.3 выше.

4.8 После получения работоспособного программного продукта приступите к итоговому оформлению пояснительной записки.

4.9 Подготовьте к защите презентацию, демонстрационный материал, устный доклад.

4.10 Доклад обычно состоит из трех основных частей:

- вступительное слово, где описывается суть поставленной задачи, актуальность и новизна решаемой проблемы (если возможно);
- основная часть доклада (подробная формулировка задачи, подходы к решению, описание выбранного способа решения, описание выполненной работы);
- заключение, то есть краткое подведение итогов (что сделано, что получено, достоинства и недостатки работы, перспективы использования работы).

На защите доклад следует рассказывать, а не читать. Ограничения по времени - не более 5 мин. Поэтому рекомендуется предварительно доклад несколько раз проговорить и оценить временные рамки и соответствие доклада и презентационного материала.

4.11 Подготовьте все необходимые материалы на электронном и бумажном носителе.

4.12 При выполнении каждого этапа работы обязательно консультируйтесь с руководителем.

4.12. Стремитесь к соблюдению графика работы, не затягивайте выполнение этапов. Это важнее, чем стремление к идеальному выполнению работы, которое иногда «затягивает» и требует очень больших затрат времени.

4.13 И, наконец, в установленный срок сдайте и защитите работу. Успехов Вам!

5 Примерный перечень информационных источников

Выполнение задания должно начинаться с исследования современных технологий, алгоритмов и методов, которые применяются в мировой практике при решении аналогичных задач. В процессе разработки программного продукта необходимую дополнительную информацию можно оперативно получать в Интернете. Студенту могут быть полезны следующие адреса информационных ресурсов:

www.intuit.ru

www.cprogramming.com

dmtsoft.ru

life-prog.ru

ЛИТЕРАТУРА

1. Вирт Н. Алгоритмы и структуры данных. Новая версия для Оберона. – Изд-во: «ДМК Пресс», 2010. – 272с. [электронный ресурс] // ЭБС «Лань» http://e.lanbook.com/books/element.php?pl1_id=1261
- 2.