

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

Бубнова Н.Д.

Введение в алгоритмы и основы технологий разработки программ

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

(на правах рукописи)
Редакция 1

Барнаул · 2015

Бубнова Н.Д. Введение в алгоритмы и основы технологий разработки программ [Электронный ресурс]: Методические указания к выполнению лабораторных работ / Н.Д. Бубнова. –Барнаул: АлтГТУ, 18 с.

Материалы практикума составлены с учетом требований ФГОС ВО по направлениям подготовки 09.03.04 «Программная инженерия» квалификация (степень) бакалавр. Отражают стандарт учебной дисциплины «Введение в алгоритмы и основы технологий разработки программ»

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

Статические структуры

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 массиве $C(n)$ подсчитать количество отрицательных и сумму положительных элементов.
2. Известно время t_1, t_2, \dots, t_n , за которое некоторую работу выполнит каждый из n рабочих бригады, работая в одиночку. Сколько времени понадобится бригаде на выполнение этой работы, если они будут работать совместно (никто не «сачкует»)?
3. Элементы массива $A(n^2)$ построчно расположить в матрице $B(n, n)$.
4. Из каждого из m чисел x_1, x_2, \dots, x_m вычесть их среднее арифметическое:

$$x_{cp} = \frac{1}{m} \sum_{i=1}^m x_i; \quad x'_i = x_i - x_{cp}, \quad i = 1, 2, \dots, m.$$

5. Вычислить среднее значение x_{cp} и дисперсию d_x для заданного массива $X(k)$ наблюдений:

$$x_{cp} = \frac{1}{k} \sum_{i=1}^k x_i;$$

$$d_x = \frac{1}{k-1} \sum_{i=1}^k (x_i - x_{cp})^2.$$

6. Вычислить среднее значение x_{cp} и дисперсию d_x для заданного массива $X(k)$ наблюдений более эффективным (по сравнению с предыдущей задачей) способом, за один проход массива:

$$s_1 = \sum_{i=1}^k x_i; \quad s_2 = \sum_{i=1}^k x_i^2; \quad x_{cp} = \frac{s_1}{k}; \quad d_x = \frac{s_2}{k-1} - \frac{s_1^2}{k(k-1)}.$$

7. Найти угол между векторами $A(n)$ и $B(n)$, используя формулу:

$$\cos \varphi = \frac{(A, B)}{|A| \cdot |B|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}}.$$

8. В матрице $Z(m, m)$ каждый элемент разделить на диагональный, стоящий в том же столбце.

9. В массиве $C(m)$ каждый третий элемент заменить полусуммой двух предыдущих, а стоящий перед ним – полусуммой соседних с ним элементов.

10. В матрице $A(m, n)$ все ненулевые элементы заменить обратными по величине и противоположными по знаку.

11. Найти среднее арифметическое элементов каждой строки матрицы $Q(l, m)$ и вычесть его из элементов этой строки.

12. Задана матрица $A(k, l)$. Найти вектор $B(l)$, каждый элемент которого равен среднему арифметическому элементов соответствующего столбца матрицы A .

13. Все ненулевые элементы матрицы $D(k, l)$ расположить в начале массива $E(k \times l)$ и подсчитать их количество.

14. Дан массив $A(n)$. Все положительные его элементы поместить в начало массива $B(n)$, а отрицательные элементы – в начало массива $C(n)$. Подсчитать количество тех и других.

15. Элементы заданного массива $T(k)$ расположить в обратном порядке: $t_k, t_{k-1}, \dots, t_2, t_1$.

16. Заданы массивы $A(m)$ и $B(n)$. Построить массив $C(m+n)$, расположив в начале его элементы массива A , а затем – элементы массива B .

17. Все четные элементы целочисленного массива $K(n)$ поместить в массив $L(n)$, а нечетные – в массив $M(n)$. Подсчитать количество тех и других.

18. В массиве $A(n)$ локальных максимумов, то есть таких a_i , что $a_{i-1} < a_i > a_{i+1}$.

19. В массиве $Z(2n)$ каждый элемент с четным индексом поменять местами с предыдущим, то есть получить последовательность $Z_2, Z_1, Z_4, Z_3, \dots, Z_{2n}, Z_{2n-1}$.

20. Многочлен $P_n(x)$ задан массивом своих коэффициентов $A(n+1)$. Найти массив коэффициентов производной этого многочлена.

21. Многочлен $P_n(x)$ задан массивом своих коэффициентов $A(n+1)$. Вычислить значение многочлена для заданного значения x (полезно использовать схему Горнера).

22. Шахматная доска. Целочисленный массив $K(n, n)$ заполнить нулями и единицами, расположив их в шахматном порядке.

23. В массиве $Z(m)$ найти число чередований знака, то есть перехода с минуса на плюс или с плюса на минус. Например, в последовательности 0, -2, 0, -10, 2, -1, 0, 0, 3, -3 четыре чередования.

24. Латинским квадратом порядка n называется квадратная таблица размером $n \times n$, каждая строка и каждый столбец которой содержит все числа от 1 до n . Для заданного n в матрице $L(n)$,

n) построить латинский квадрат порядка n.

25. Нарастающий итог. В массиве A(n) каждый элемент, кроме первого, заменить суммой всех предыдущих элементов.

26. Заполнить матрицу M(k, l) числами 1, 2, 3, 4 так, чтобы по горизонтали, вертикали и диагонали не было одинаковых рядом стоящих чисел.

27. На шахматной доске находятся король и несколько ферзей другого цвета. Проверить, находится ли король под боем и если да, кто ему угрожает. Положение фигур задано массивом K(8,8): 0 - клетка пуста, 1 - король, 2 - ферзь. Ферзь бьет по горизонтали, вертикали и диагоналям.

28. Результаты сессии, состоящей из трех экзаменов, для группы из n студентов представлены матрицей K(n,3). Оценка ставится по четырехбальной системе; неявка обозначена единицей. Подсчитать количество неявок, неудовлетворительных, удовлетворительных, хороших и отличных оценок по каждому экзамену.

29. Текущее сглаживание. Каждый из элементов x_i массива X(n) заменить средним значением первых i элементов этого массива.

30. Каждый из элементов t_i массива T(m) заменить минимальным среди первых i элементов этого массива.

31. 4Турнирная таблица. В матрице K(n,n) представлена турнирная таблица соревнований

по футболу среди n участников (каждый элемент a_{ij} матрицы есть число голов, забитых i-м участником i-му участнику); все элементы главной диагонали равны нулю. Присвоить каждому диагональному элементу разницу забитых и пропущенных голов, то есть разность между суммами элементов соответствующих строки и столбца.

32. 4.31. (8 б.) Осуществить циклический сдвиг элементов массива T(n) на m позиций

влево, то есть получить массив: $t_{m+1}, \dots, t_n, t_1, \dots, t_m$. При этом не обязательно $m < n$.

33. Просуммировать элементы матрицы A(n,n) по каждой из линий, параллельных главной диагонали. Напечатать полученные суммы.

34. Для заданной матрицы A(m,n) найти ее норму:

$$\|A\| = \max_{i=1,m} \sum_{k=1}^n |a_{ik}|$$

35. Расписание. Известно время начала (например, 6:00) и окончания (24:00) работы некоторого пригородного автобусного маршрута с несколькими (n) автобусами на линии, а также протяженность маршрута в минутах (в один конец) и время отдыха на конечных остановках. Составить суточное расписание этого маршрута (моменты отправления с конечных пунктов) без учета обеда и пересмен.

Лабораторная работа №2

Полустатические структуры

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. Написать программу для вычисления значения арифметического выражения по его обратной польской записи, если допустима унарная операция замены знака
2. Написать программу преобразования арифметического выражения в обратную польскую запись.
3. Поступают шарики четырех цветов, которые распределяются между тремя списками. Как только в конце всех списков окажутся шарики одного цвета, они удаляются. Подсчитать, сколько раз выполнялось удаление, если поступило 150 шариков.
4. Сравнить на следование в одинаковом порядке одинаковых элементов от начала к концу магазина и слева направо в деке.
5. Сравнить на следование заданного числа одинаковых элементов от начала к концу стека и справа налево в деке.
6. Сравнить на следование в одинаковом порядке одинаковых элементов в магазине и деке.
7. Приписать все элементы магазина с учетом порядка их поступления к правому концу дека.
8. Написать программу для чтения левого элемента из дека, если в верхушке исходного магазина содержится отрицательный элемент и правого элемента из дека в противном случае.
9. Получить стек, каждый элемент которого является суммой соответствующих элементов двух исходных стеков.
10. Написать процедуру для получения очереди, каждый элемент которой является суммой соответствующих элементов магазина и очереди.
11. Разбить стек на 2 стека, отнеся к первому те элементы, на номера которых указывает исходная очередь.
12. Удалить из магазина все элементы, поступившие после элемента с заданным значением.
13. Слить две очереди в одну.
14. Выбрать из исходного дека элементы, на номера которых указывает магазин, и поместить их в стек. Нумерация элементов дека выполняется справа.
15. Выбрать из исходного дека элементы, на номера которых указывает магазин, и поместить их в стек. Нумерация элементов дека выполняется слева.
16. Получить магазин из минимальных соответствующих элементов исходных деков, просматриваемых слева направо.
17. Получить очередь из минимальных соответствующих элементов исходных магазинов.
18. Разбить очередь на две, включив в первую первые 10 элементов исходной.
19. Заданы два стека и очередь. Получить магазин, включая в него суммы соответствующих элементов стеков в том порядке, в каком они перечислены в очереди.
20. Исключить из стека столько элементов, сколько отрицательных значений в исходной очереди.
21. Задано число. Если оно положительно, исключение выполняется для правого конца дека, если отрицательно - из левого. На число исключаемых элементов указывает модуль исходного числа.
22. Написать программу для получения стека, элементы которого получаются как сумма соответствующих, указанных очередью, элементов двух исходных магазинов.

23. Сравнить магазин и очередь на эквивалентность.
24. Написать программу для слияния двух очередей, содержащих информацию об ожидающих получения квартир (ФИО, адрес, численность семьи, дата поступления в очередь), в одну очередь, упорядоченную в соответствии с датой поступления в нее.
25. Гаражная стоянка имеет одну стояночную полосу и может разместить до 10 машин. Въезд и выезд на одном конце полосы. Если владелец хочет забрать машину, не являющуюся последней, то машины, стоящие за ней, удаляются, машина выводится и удаленные машины возвращаются в исходном порядке на место. Реализовать заданную последовательность операций на стоянке. Формат операции: <вид операции> <регистрационный номер машины>
<вид операции>: А - поставить
Д - удалить
26. Программа печатает сообщение о прибытии или отъезде любой машины. Информация о переполнении должна выдаваться. При выезде машины со стоянки должна выдаваться информация о том, сколько раз машина удалялась со стоянки для обеспечения выезда других автомобилей.
27. Написать программу, имеет ли вводимая символьная строка формат `aDbDcD...Dz`, где каждая строка `a,b,...,z` имеет формат `xСу`, где `x` состоит из символов `a` и `b`, а `y` - обратная `x` строка. При чтении можно считывать только следующий символ строки.
28. Написать программу, имеет ли вводимая символьная строка формат `xСу`, где `x` состоит из символов `A` и `B`, а `y` - обратная `x` строка. При чтении можно считывать только следующий символ строки.

Лабораторная работа №3

Односвязные списки

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. Написать программу для умножения двух полиномов.
2. Написать программу для нахождения остатка от деления двух полиномов.
3. Задан список целых чисел. Образовать новый, в который включаются все простые числа исходного списка.
4. Задан список чисел. Образовать из него новый, исключив из исходного минимальное число элементов так, чтобы список стал неубывающим.
5. Задан список. Изменить в нем все связи на обратные.
6. Задан список строк. Исключить из списка одинаковые строки.
7. Задан лексикографически упорядоченный список строк. Добавить в список новую строку так, чтобы список остался упорядоченным.
8. Задан список абитуриентов. Создать из него список медалистов.

9. Задан список абитуриентов. Создать из него список выпускников городских школ
10. Задан список абитуриентов. Создать из него список для зачисления.
11. Задан список абитуриентов. Удалить из него тех абитуриентов, которые не представили подлинник аттестата.
12. Заданы списки абитуриентов, поступающих на ПОВТ и ИП. Выделить в отдельный список абитуриентов, включенных в оба списка.
13. Задан список студентов. Исключить из списка отчисленных студентов, информация о которых содержится в файле.
14. Перевести из списка студентов 3 курса в список студентов 2 курса всех ушедших в академический отпуск. Информация о них содержится в файле.
15. Заданы два списка. Найти в них общие подсписки максимальной длины.
16. Заданы два списка. Пополнить первый уникальными элементами второго.
17. Заданы два списка. Пополнить первый неуникальными элементами второго, отсутствующими в первом списке.
18. Заданы три списка. Создать новый список из элементов, встречающихся во всех списках.
19. Заданы три списка. Создать новый список из элементов, встречающихся в первом и отсутствующих в остальных списках.
20. Создать новый список из элементов трех упорядоченных списков так, чтобы получился упорядоченный список (выполнить слияние списков).
21. Разбить список чисел на подсписки так, чтобы в одном были элементы кратные 3, в другом четные элементы. Элементы, не удовлетворяющие сформулированным критериям, оставить в исходном списке.
22. В очередь поступают задания, характеризующиеся именем и приоритетом. Организовать работу с очередью, если поступающее в очередь задание встанет последним из заданий с тем же приоритетом. Очередь упорядочена по убыванию приоритетов.
23. Разбить список на два подсписка так, чтобы элементы распределились между ними поровну, если это возможно.
24. Из двух списков образовать список уникальных элементов.
25. Из списка исключить положительные элементы
26. Отрицательные элементы списка переместить в его конец.
27. Написать программу для упорядочения односвязного списка.
28. Положительные элементы списка переместить в его начало.

Лабораторная работа №4

Двухсвязные списки

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 среди элементов списка наибольший и наименьший элементы.
2. Связать в новый список все четные числа из списка положительных чисел.

3. В двух списках хранится исходная строка символов и строка для замены. Каждый символ находится в отдельном элементе списка. Необходимо удалить из исходной строки символы с i по j , а затем вставить в исходную строку, начиная с i -того символа, строку для замены. Длина строки для замены может не совпадать с длиной удаленной части исходной строки
4. Включить новый элемент перед i -тым в списке.
5. Написать программу дополнения упорядоченного списка новым элементом без нарушения упорядоченности.
6. Слить два списка в один упорядоченный по возрастанию список.
7. Вставить второй список вместо i -того элемента первого списка.
8. Вставить второй список перед i -тым элементом первого списка.
9. Проверить, является ли второй список подсписком первого.
10. Разбить список на 2 подсписка: к первому подписку отнести подсписок заданной длины, начиная с i -того элемента списка. Остальные элементы отнести ко второму подписку.
11. Подсчитать, сколько раз заданный список встречается в исходном списке.
12. Подсчитать, сколько раз встречается в исходном списке заданный элемент.
13. Переставить местами элементы, обрамляющие i -тый элемент списка.
14. Заменить все отрицательные элементы списка их квадратами.
15. Во входном потоке задано некоторое число элементов. Построить алгоритм формирования двусвязного списка из этих элементов.
16. Из неупорядоченного списка исключить все элементы, следующие за элементом с заданным значением.
17. Разбить исходный список на 2: к первому отнести все положительные элементы, ко второму - отрицательные.
18. Исключить все элементы с заданным значением из списка.
19. Сравнить два списка на идентичность.
20. Вставить вместо элемента с заданным значением новый список.
21. Напечатать значения элементов списка, предшествующих элементам с положительным значением.
22. Подсчитать число элементов, совпадающих по значению с переменной T .
23. Разбить список на два одинаковых подсписка (по числу элементов).
24. Из упорядоченного списка исключить все элементы, принадлежащие $[a, b]$.
25. Разбить упорядоченный список на два: к первому отнести элементы из $[a, b]$, ко второму - все остальные.
26. 86. В односвязном списке найти элементы с минимальным значением. Удалить все элементы, заключенные между самым левым и самым правым вхождением этого элемента в список.
27. Написать программу для корректировки элементов первого списка информацией из второго. Создать третий список, дублирующий те элементы первого, которых нет во втором.
28. Найти из элементов двухсвязного списка элемент с минимальным значением. Если этот элемент меньше заданного значения, исключить из списка элементы, предшествующие самому левому его вхождению в список.
29. Найти из максимальный элемент списка. Если он больше заданного значения, то исключить из списка элементы, следующие за самым правым вхождением максимального элемента.

Лабораторная работа №5

Циклические односвязные и двухсвязные списки с головой и без головы

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. Напечатать все элементы упорядоченного списка, принадлежащие $[a, b]$. Значения a и b задаются.
2. Напечатать все элементы упорядоченного списка, не принадлежащие $[a, b]$. Значения a и b задаются.
3. Присоединить все элементы упорядоченного списка, принадлежащие $[a, b]$ к его концу. Значения a и b задаются.
4. Даны односвязный циклический список и массив значений элементов для замены. Просмотреть исходный список и заменить 1 отрицательный элемент списка 1 элементом массива, 2 отрицательный элемент - 2 элементом массива и т.д.
5. Задан двусвязный циклический список с головой. Напечатать его. Удалить 1 отрицательный элемент этого списка. Напечатать результирующий список.
6. Проверить, содержится ли второй двусвязный список в первом, и если да, то все элементы, предшествующие в первом списке второму, выделить в третий список.
7. Написать программу для создания из элементов трех исходных связных списков с головой нового списка, содержащего элементы, принадлежащие 1 и 3, но не принадлежащие 2.
8. Заданы два списка, каждый из которых представляет в памяти арифметическое выражение. Найти их сумму и привести подобные.
9. Разбить исходный односвязный циклический список на 2 подсписка так, чтобы количества элементов в них были почти одинаковы, и чтобы ключевое поле любого элемента первого подписка было больше значения ключевого поля любого элемента второго подписка.
10. Из двух связных списков образовать один по правилу: взять из 1 и 2 списков элементы, содержащие данные об одном объекте, и занести расширенные сведения об этом объекте в результирующий список.
11. Дополнить упорядоченный односвязный циклический список новой информацией так, чтобы список остался упорядоченным и сделать этот список списком с головой.
12. Удалить из односвязного циклического списка элементы, ключевые поля которых не принадлежат заданному диапазону.
13. Удалить из двухсвязного списка с головой элементы, ключевые поля которых принадлежат заданному диапазону.
14. Написать программу для исключения из 2 списка элементов со значением ключевого поля, большим среднего значения ключевого поля всех элементов первого списка.

15. Написать программу для получения из односвязного, двухсвязного и двухсвязного списка с головой циклического списка уникальных элементов.
16. Заданы два списка. Образовать третий список, содержащий все элементы второго списка, не принадлежащие первому.
17. Заданы два списка. Разбить первый на два подсписка так, чтобы к первому подписку относились элементы со значением ключевого поля, не совпадающим со значением ключевого поля ни одного элемента второго списка.
18. Заданы два списка с одинаковым числом элементов. Разбить первый список на два подсписка, отнеся к первому подписку элементы со значением, меньшим значения соответствующего элемента второго списка.
19. Найти одинаковые элементы в односвязном и двухсвязном списках и исключить их из обоих.
20. Заданы двухсвязный и односвязный списки. Включить элементы второго списка слева от элемента с заданным значением 1 списка.
21. Написать программу для слияния упорядоченных односвязного и линейного списка, отображенного в векторную память, так, чтобы результирующий двухсвязный список был упорядоченным.
22. Задан односвязный циклический список с головой. Написать программу для нахождения максимального числа и удаления всех элементов, содержащихся между самым правым элементом с максимальным значением и головой списка.
23. Заданы два односвязных неупорядоченных списка с головой. Объединить элементы, входящие в них так, чтобы они образовали один упорядоченный список.
24. Заданы два двухсвязных циклических списка с головой. Объединить их так, чтобы элементы с одним значением ключевого поля встречались в объединенном списке не более одного раза.
25. Задан двухсвязный неупорядоченный список. Написать программу для его упорядочения.
26. Заданы два двухсвязных циклических списка. Элементы первого упорядочены по возрастанию, элементы второго - по убыванию. Исключить из второго списка элементы, встречающиеся в первом.
27. Создать список, упорядоченный по приоритету. Откорректировать список на основе введенной информации. Виды корректировки:
 - а) изменить содержимое одного или нескольких информационных полей;
 - б) удалить элемент;
 - в) добавить элемент.
28. Найти максимальное значение в односвязном циклическом списке с головой. Вставить между двумя соседними вхождениями максимального значения другой список, начиная с самого левого элемента.

Лабораторная работа №6

Бинарные 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. Проверить, содержится ли второе дерево как поддереву в первом дереве, и если содержится, определить уровень, на котором находится в 1 дереве узел, соответствующий корню второго дерева.
2. Дерево представлено в памяти в виде бинарного дерева. Определить максимальное число потомков одного узла в исходном дереве.
3. Удалить вершину с заданным значением ключа из бинарного дерева поиска.
4. Два бинарных дерева проверить на эквивалентность.
5. Занести в каждый узел бинарного дерева номер его уровня.
7. Восстановить неупорядоченное бинарное дерево по известным порядкам его обходов сверху вниз и слева-направо.
8. Найти высоту бинарного дерева.
9. Заданы 2 бинарных дерева. Удалить из второго бинарного дерева информацию, содержащуюся в обоих деревьях.
10. Получить копию бинарного дерева
11. Найти в бинарном дереве узел с заданным значением ключа и удалить его правого потомка.
12. Подсчитать число листьев в бинарном дереве.
13. Добавить во второе бинарное дерево те узлы первого, которые во втором ранее не содержались.
14. Представить заданное арифметическое выражение в виде БД. (Допустимы +, -, /, *, **, скобки; имя переменной - одна буква).
15. Обойти бинарное дерево в прямом порядке.
16. Построить процедуру получения всех элементов, размещенных на дереве сортировок с признаками в заданном диапазоне.
17. Построить процедуру определения среднего числа сравнений для доступа к любому элементу заданного дерева.
18. Получить скобочное представление бинарного дерева.
19. Каталог организован в виде упорядоченного БД. Каждый узел соответствует файлу и содержит имя файла, дату последнего обращения. Написать программу, которая обходит дерево и удаляет все файлы, обращение к которым происходило до определенной даты.
20. Написать процедуру поиска непосредственно предшествующего элемента.
21. Реализовать алгоритм получения упорядоченной последовательности элементов, рассортированных на дереве, а затем построения выравненного дерева на этой последовательности.
22. В дереве частота обращений к каждому узлу измеряется эмпирически - приписыванием узлу счетчика обращений. Через определенный интервал дерево реорганизуется и узлы расставляются по убыванию счетчиков частоты обращений. Выполнить реорганизацию дерева.
23. Написать процедуру поиска непосредственно следующего элемента.
24. Реализовать добавление элемента в дерево, прошитое для обхода по правилу: корень, левое поддерево, правое поддерево.
25. Реализовать добавление элемента в дерево, прошитое для обхода по правилу: левое поддерево, корень, правое поддерево.
26. Реализовать удаление элемента из дерева, прошитого для обхода по правилу: левое поддерево, корень, правое поддерево.

Лабораторная работа №7

АВЛ-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. Обойти АВЛ дерево сверху вниз и удалить запись с заданным для этого обхода номером.
2. Обойти АВЛ дерево слева направо и удалить запись с заданным для этого обхода номером.
3. Обойти АВЛ дерево справа налево и удалить запись с заданным для этого обхода номером.
4. Удалить из исходного дерева запись с заданным значением ключа.
5. Сравнить на информационную эквивалентность два АВЛ-дерева.
6. Проверить, является ли первое АВЛ-дерево поддеревом второго.
7. Проверить, все ли ключи второго АВЛ-дерева содержатся в первом.
8. Удалить из АВЛ-дерева ключи, принадлежащие заданному диапазону значений.
9. Удалить из второго АВЛ-дерева ключи, содержащиеся в первом.
10. Построить АВЛ-дерево из записей с такими ключами второго, которые не содержатся в первом.
11. Определить процент ключей первого АВЛ-дерева порядка n , принадлежащих второму АВЛ-дереву.
12. Удалить из второго АВЛ-дерева записи с ключами, принадлежащими первому АВЛ-дереву..
13. Определить в АВЛ-дереве след заданного ключа и удалить его, если последняя цифра следа 0.
14. Удалить ключ с заданным следом из АВЛ-дерева.
15. Для заданного ключа найти в АВЛ-дереве предшественника и последователя и исключить их.
16. Подсчитать среднее количество разбалансированных узлов на каждом уровне АВЛ-дерева. Исключить все разбалансированные записи уровня, количество разбалансированных записей которого максимально отклоняется от найденного среднего.
17. Подсчитать число записей в АВЛ-дереве. Предложить такой порядок поступления записей, чтобы высота полученного дерева была бы минимальной.
18. Задана очередь, каждый элемент которой содержит вид операции над деревом и необходимую для ее выполнения информацию.
Выполнить операции в АВЛ-дереве в соответствии с информацией, содержащейся в очереди.
19. Упорядочить записи в АВЛ-дереве по величине задолженности абонентов. О каждом абоненте в АВЛ-дереве содержатся следующие данные: величина задолженности, номер телефона, фамилия абонента, адрес. Напечатать списки должников по оплате и реализовать процедуры корректировки дерева.
20. Задано АВЛ-дерево и некоторое число. Разбить исходное дерево на два так, чтобы информация с ключевым полем большим заданного числа перешла бы в новое дерево, а остальные данные остались бы в исходном.
21. Указать, какому диапазону должен принадлежать ключ, добавление которого в АВЛ-дерево вызовет минимальную перестройку дерева.

22. Проверить на эквивалентность два AVL-дерева. Если они не являются информационно эквивалентными, указать минимальное по мощности множество ключей которое надо удалить, чтобы деревья стали эквивалентными.
23. Проверить на эквивалентность два AVL-дерева. Если они не эквивалентны, указать минимальное по мощности множество ключей которое надо добавить, чтобы деревья стали эквивалентными. Указать каким диапазонам значений должны принадлежать ключи и сколько ключей надо взять из каждого диапазона значений.
24. Определить, сколько ключей можно добавить в AVL-дерево так, чтобы его высота не изменилась.
25. Определить, как изменится высота дерева после добавления в него заданных записей.

Лабораторная работа №8

Декартовы 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. Подсчитать, сколько чисел на отрезке с l по r лежат в интервале от a до b .
2. Удалить из декартова дерева узлы с ключами, принадлежащими отрезку с l по r .
3. Подсчитать сумму чисел на отрезке с l по r .
4. Найти максимум на отрезке с l по r .
5. Найти минимум на отрезке с l по r .
6. Общие записи 2 декартовых деревьев выделить в отдельное дерево с удалением из исходных.
7. Пополнить 1-ое декартово дерево записями второго.
8. Заданы 2 декартовых дерева. Из дерева, максимальный элемент которого больше, удалить записи, принадлежащие заданному отрезку.
9. Заданы 2 декартовых дерева. Из дерева, минимальный элемент которого меньше, удалить записи, принадлежащие заданному отрезку.
10. Пополнить 1 декартово дерево записями 2-го, принадлежащими заданному отрезку.
11. Объединить в одно дерево записи 1 дерева, принадлежащие первому отрезку, и записи 2 дерева, принадлежащие второму отрезку.
12. Переставить заданную часть массива после заданного элемента.
13. Увеличить все элементы заданного отрезка в 2 раза.
14. Найти общего предка элементов, принадлежащих 2 заданным диапазонам
15. Удалить элемент с заданным номером.
16. Вычислить сумму элементов с приоритетами из заданного диапазона значений.
17. Подсчитать количество элементов декартова дерева с ключами из заданного диапазона значений.
18. Подсчитать количество и сумму всех элементов декартова дерева.
19. Изменить элемент с заданным значением ключа, увеличив значение ключа в два раза.

20. Подсчитать количество элементов декартова дерева с ключами из заданного диапазона значений и увеличить значения выявленных ключей на заданное значение
21. . Уменьшить ключи элементов декартова дерева со значениями приоритетов из заданного диапазона на заданное значение
22. Подсчитать количество элементов декартова дерева с приоритетами, превышающими приоритет элемента с заданным значением ключа
23. Удалить из декартова дерева узел со значением ключа, минимально отличающимся от заданного значения.

Лабораторная работа №9

Работа с таблицами

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. Преобразовать последовательную таблицу в древовидную. Элементы из последовательной таблицы выбирать так, чтобы сформировалось выровненное дерево.
2. Процедура поиска, включения и исключения для древовидных таблиц, организованных в векторной памяти.
3. Построить хеш-копию таблицы, организованной как бинарное дерево поиска.
4. Откорректировать таблицу, организованную как бинарное дерево поиска, на основе последовательной таблицы.
5. Перейти от хеш-таблицы к древовидной таблице.
6. Сравнить на эквивалентность хеш-таблицу с последовательной таблицей.
7. Откорректировать таблицу с разрешением коллизий через связанные списки на основе таблицы, организованной как бинарное дерево поиска.
8. Проверить на эквивалентность таблицу, организованную как бинарное дерево поиска, и таблицу с разрешением коллизий через связанные списки.
9. Процедура занесения и поиска записи в гибридной таблице, если вступившие в коллизию записи связаны в список.
10. Пополнить хеш-таблицу информацией из бинарного дерева поиска.
11. Перейти от таблицы с разрешением коллизий через бинарное дерево поиска к последовательной таблице.
12. Дополнить таблицу- бинарное дерево поиска информацией из хеш-таблицы.
13. Перейти от таблицы с разрешением коллизий через связанные списки к последовательной таблице.
14. Перейти от таблицы- бинарное дерево поиска, к последовательной упорядоченной таблице.
15. Откорректировать таблицу- бинарное дерево поиска на основе последовательной таблицы.
16. Выделить в отдельную таблицу- бинарное дерево поиска те записи исходной таблицы- бинарное дерево поиска, которые не содержатся в исходной хеш-таблице.
17. Сравнить хеш-таблицу и древовидную (бинарное дерево поиска) на эквивалентность.

18. Выделить в отдельную таблицу- бинарное дерево поиска те записи исходной таблицы- бинарное дерево поиска, которые содержатся в исходной хеш-таблице.
19. Написать процедуру для удаления информации из хеш-таблицы и занесения информации в нее.
20. Задана таблица с коллизиями, разрешенными через бинарное дерево поиска. Построить последовательную таблицу и связать в список ту ее часть, которая содержит равные значения ключей.
21. Построить хеш-таблицу, являющуюся копией последовательной неупорядоченной таблицы.
22. Построить процедуры размещения и поиска записей в таблице с коллизиями, если для поиска нужной позиции просматриваются последовательно все позиции, начиная с первой, вычисляемой с помощью заданной функции расстановки.
23. Таблица, представляющая собой бинарное дерево поиска, отображена в векторную память. Реализовать алгоритм удаления из таблицы одной записи.
24. Заданы таблица расстановок и таблица переполнения, отображенные в векторную память. Реализовать процедуры занесения информации в такую таблицу и удаления информации из нее.
25. Откорректировать последовательную таблицу на основе таблицы с разрешением коллизий на основе ДДП.

Лабораторная работа № 10

Разреженные матрицы

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. Заданы две матрицы (А - трехдиагональная и В - пятидиагональная) и вектор X. Выполнить следующие действия: $(A+B)*X$
2. Заданы две матрицы (А-трехдиагональная и В-нижняя треугольная) и вектор X. Выполнить следующие действия: $(A+c*B)*X$, где c – число.
3. Заданы две матрицы (А - трехдиагональная и В - верхняя треугольная) и вектор X. Выполнить следующие действия: $(c*A+B)*X$, где c – число.
4. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная методом логических шкал) и вектор X. Выполнить следующие действия: $c*(A+B)*X$, где c – число.
5. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная модифицированным методом логических шкал) и вектор X. Выполнить следующие действия: $(c*A+B)*X$, где c – число.
6. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная методом пар) и вектор X. Выполнить следующие действия: $(A+c*B)*X$, где c – число.
7. Заданы две матрицы (А - трехдиагональная и В - блочная матрица) и вектор X. Выполнить следующие действия: $(A+c*B)*X$, где c – число.

8. Заданы две матрицы (А-пятидиагональная и В-нижняя треугольная) и вектор Х. Выполнить следующие действия: $(A+c*B)*X$, где с – число.
9. Заданы две матрицы (А - пятидиагональная и В - верхняя треугольная) и вектор Х. Выполнить следующие действия: $(c*A+B)*X$, где с – число.
10. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная методом логических шкал) и вектор Х. Выполнить следующие действия: $c*(A+B)*X$, где с – число.
11. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная модифицированным методом логических шкал) и вектор Х. Выполнить следующие действия: $(c*A+B)*X$, где с – число.
12. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная методом пар) и вектор Х. Выполнить следующие действия: $(A+c*B)*X$, где с – число.
13. Заданы две матрицы (А - пятидиагональная и В - блочная матрица) и вектор Х. Выполнить следующие действия: $(A+c*B)*X$, где с – число.
14. Заданы две матрицы (А - пятидиагональная и В - блочная матрица). Перемножить эти матрицы.
15. Заданы две матрицы (А - трехдиагональная и В - блочная матрица). Перемножить эти матрицы.
16. Заданы две матрицы (А - трехдиагональная и В - пятидиагональная). Перемножить эти матрицы.
17. Заданы две матрицы (А-трехдиагональная и В-нижняя треугольная). Перемножить эти матрицы.
18. Заданы две матрицы (А - трехдиагональная и В - верхняя треугольная) Перемножить эти матрицы.
19. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная методом логических шкал) Перемножить эти матрицы.
20. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная модифицированным методом логических шкал) и вектор Х Перемножить эти матрицы.
21. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная методом пар) Перемножить эти матрицы.
22. Заданы две матрицы (А - трехдиагональная и В - блочная матрица) и вектор Х. Перемножить эти матрицы.
23. Заданы две матрицы (А-пятидиагональная и В-нижняя треугольная) и вектор Х. Перемножить эти матрицы.
24. Заданы две матрицы (А - пятидиагональная и В - верхняя треугольная) и вектор Х. Перемножить эти матрицы.
25. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная методом логических шкал). Перемножить эти матрицы.
26. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная модифицированным методом логических шкал) Перемножить эти матрицы.
27. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная методом пар) Перемножить эти матрицы.
28. Заданы две матрицы (А - пятидиагональная и В - блочная матрица) Перемножить эти матрицы.
29. Заданы две матрицы (А - пятидиагональная и В - блочная матрица). Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A-c*B$, где с – число.
30. Заданы две матрицы (А-трехдиагональная и В-нижняя треугольная) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A-c*B$, где с – число.

31. Заданы две матрицы (А - трехдиагональная и В - верхняя треугольная) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия:
32. $A - c * B$, где c – число.
33. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная методом логических шкал) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.
34. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная модифицированным методом логических шкал) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.
35. Заданы две матрицы (А - трехдиагональная и В - разреженная матрица, уплотненная методом пар) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.
36. Заданы две матрицы (А - трехдиагональная и В - блочная матрица) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $tA - c * B$, где c – число.
37. Заданы две матрицы (А - пятидиагональная и В - нижняя треугольная) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.
38. Заданы две матрицы (А - пятидиагональная и В - верхняя треугольная) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.
39. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная методом логических шкал) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.
40. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная модифицированным методом логических шкал) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.
41. Заданы две матрицы (А - пятидиагональная и В - разреженная матрица, уплотненная методом пар) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.
42. Заданы две матрицы (А - пятидиагональная и В - блочная матрица) Перейти от их уплотненного представления в памяти к обычному и выполнить следующие действия: $A - c * B$, где c – число.

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