

# МЕТОДЫ СОРТИРОВКИ И ПОИСКА ДАННЫХ

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 1

Тема: Метод прямого выбора.

Цель работы: Изучение методов квадратичной сортировки массивов.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить ручную сортировку методом прямого выбора. При тестировании программы можно использовать данный пример для проверки правильности реализации алгоритма.
2. Разработать подпрограмму сортировки массива методом прямого выбора. Правильность сортировки проверить путем подсчета контрольной суммы и числа серий в массиве. Предусмотреть подсчет фактического количества пересылок и сравнений ( $M$  и  $C$ ) и сравнить их с теоретическими оценками для различных типов массивов.
3. Экспериментально определить среднюю длину серии в случайном массиве.\*
4. Придумать способ устранения фиктивных перестановок и оценить его влияние на фактические значения  $M$  и  $C$  для метода прямого выбора.\*
5. Исследовать метод прямого выбора на массивах убывающих, возрастающих и случайных чисел и сделать вывод о зависимости (или независимости) метода от исходной упорядоченности массива.\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 2

Тема: Метод пузырьковой сортировки и шейкерная сортировка.

Цель работы: Изучение методов квадратичной сортировки массивов.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить ручную сортировку методом пузырьковой сортировки и шейкерную сортировку. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.
2. Разработать подпрограммы для пузырьковой и шейкерной сортировок массива целых чисел по возрастанию.
3. Предусмотреть подсчет фактического количества пересылок и сравнений ( $M_{\phi}$  и  $C_{\phi}$ ), сравнить с теоретическими оценками  $M$  и  $C$ .
4. Сравнить время работы пузырьковой и шейкерной сортировок на массивах убывающих, возрастающих и случайных чисел (по сумме  $M+C$ ). Составить таблицу вида:

Таблица 1

Размер массива	$M_{\phi}+C_{\phi}$ пузырь.			$M_{\phi}+C_{\phi}$ шейкер.		
	Убыв.	Случ.	Возр.	Убыв.	Случ.	Возр.
100						
200						
300						
400						
500						

5. Построить на экране в одной координатной плоскости графики зависимости  $M_{\phi}+C_{\phi}$  от длины массива для пузырьковой и шейкерной сортировок (для массива случайных чисел). Сравнить полученные графики зависимостей с графиком квадратичной функции. \*

### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3

Тема: Метод прямого включения и метод Шелла.

Цель работы: Изучение методов квадратичной сортировки массивов.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить ручную сортировку методом прямого включения и методом Шелла. При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.
2. Реализовать программно сортировку массива целых чисел методом прямого включения. Сравнить время работы методов квадратичной трудоемкости. Построить таблицу и проанализировать полученные результаты (использовать результаты предыдущих лабораторных работ):

Таблица 2

Длина массива	$M_{\phi}+C_{\phi}$ Метод прямого выбора	$M_{\phi}+C_{\phi}$ Метод пузырьковой сортировки	$M_{\phi}+C_{\phi}$ Метод шейкерной сортировки	$M_{\phi}+C_{\phi}$ Метод прямого включения
100				
200				
300				
400				
500				

3. Разработать подпрограмму сортировки массива целых чисел методом Шелла.
4. Исследовать трудоемкость метода Шелла для  $n=10, 100, \dots, 500$ ,  $n$  – количество элементов в массиве. Определить последовательность шагов для предварительных сортировок по формуле Кнута. Построить таблицу и проанализировать полученные результаты:

Таблица 3

Длина массива	Количество шагов по формуле Кнута	Последовательность шагов по формуле Кнута	$M_{\phi}+C_{\phi}$ Метод Шелла

5. Построить на экране в одной координатной плоскости графики зависимости трудоемкости ( $M_{\phi}+C_{\phi}$ ) от длины массива для всех методов квадратичной трудоемкости (для массива случайных чисел). Сравнить полученные графики зависимостей с графиком квадратичной функции. \*
6. Исследовать метод Шелла на зависимость трудоемкости от выбора последовательности шагов. Подтвердить рекомендацию Кнута или предложить лучший вариант выбора последовательности шагов.\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4

Тема: Двоичный поиск

Цель работы: Изучение метода быстрого поиска в массиве.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить вручную быстрый поиск (двумя версиями) первой буквы имени и буквы «Я». При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.
2. Разработать подпрограммы для двоичного поиска заданного элемента в массиве (две версии), ключ поиска передается в подпрограмму в качестве параметра.
3. Сравнить трудоемкости (фактическое количество сравнений  $C_{\phi}$ ) двух версий двоичного поиска в упорядоченном массиве. Построить таблицу и проанализировать полученные результаты:

Таблица 4

Длина массива	$C_{\phi}$ I версия	$C_{\phi}$ II версия
100		
200		
...		
500		
...		
1000		

4. Реализовать программно поиск в массиве всех элементов с заданным ключом (две версии), ключ поиска передается в подпрограмму в качестве параметра.
5. Построить графики зависимости  $C_{\phi}$  от длины массива для двух версий поиска всех элементов с заданным ключом и сравнить построенные графики с графиком логарифмической функции.\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 5

### Тема: Хеширование

Цель работы: Изучение возможностей хеширования данных для организации поиска.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить хеширование вручную методом прямого связывания (размер хеш-таблицы равен 5) и методом открытой адресации (линейные и квадратичные пробы, размер хеш-таблицы равен 11). При тестировании программы можно использовать данные примеры для проверки правильности реализации алгоритмов.
2. Разработать подпрограмму хеширования массива целых чисел методом прямого связывания и подпрограмму поиска в хеш-таблице элемента по заданному ключу. Вывести на экран построенную хеш-таблицу.
3. Реализовать подпрограмму хеширования массива целых чисел методом открытой адресации. Для разрешения коллизий использовать линейные и квадратичные пробы. Вывести на экран заполненные хеш-таблицы для  $m=11$  в виде

Таблица 5 Содержимое хеш-таблицы

Номер ячейки	0	1	2	3	...			...	$m-1$
Число									

4. Подсчитать и сравнить количество коллизий при линейных и квадратичных пробах. Построить таблицу и проанализировать полученные результаты:

Таблица 6

Размер хеш-таблицы	Количество исходных чисел	Количество коллизий	
		Линейные пробы	Квадратичные пробы
13	15		
29	30		
43	45		
67	70		
83	85		

5. Организовать поиск элемента с заданным ключом для метода открытой адресации (линейные и квадратичные пробы). \*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 6

### Тема: Метод пирамидальной сортировки.

Цель работы: Изучение методов быстрой сортировки массивов.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить ручную сортировку методом пирамидальной сортировки. При тестировании программы можно использовать данный пример для проверки правильности реализации алгоритма.
2. Разработать подпрограмму пирамидальной сортировки массива целых чисел.
3. Проверить работу метода на массивах убывающих, возрастающих и случайных чисел и сделать вывод о зависимости или независимости метода от исходной упорядоченности массива. Построить таблицу и проанализировать полученные результаты:

Таблица 7

Длина массива	(M <sub>ф</sub> +C <sub>ф</sub> ) метод пирамидальной сортировки		
	Возрастающие числа	Убывающие числа	Случайные числа
100			
200			
300			
400			
500			

4. Построить на экране в одной координатной плоскости графики зависимости трудоемкости (M<sub>ф</sub>+C<sub>ф</sub>) от  $n$  ( $n$  – длина массива) для метода Шелла и пирамидальной сортировки (для массива случайных чисел) и сравнить построенные графики с графиком функции  $n \log n$ . \*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 7

Тема: Метод Хоара

Цель работы: Изучение методов быстрой сортировки массивов.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить ручную сортировку методом Хоара. При тестировании программы можно использовать данный пример для проверки правильности реализации алгоритма.
2. Разработать подпрограмму сортировки массива целых чисел методом Хоара (две версии).
3. Проверить работу метода Хоара на массивах убывающих, возрастающих и случайных чисел и сделать вывод о зависимости или независимости метода от исходной упорядоченности массива. Построить таблицу и проанализировать полученные результаты:

Таблица 8

Длина массива	(M <sub>ф</sub> +C <sub>ф</sub> ) метод Хоара		
	Возрастающие числа	Убывающие числа	Случайные числа
100			
200			
300			
400			
500			

4. Построить на экране в одной координатной плоскости графики зависимости трудоемкости (M<sub>ф</sub>+C<sub>ф</sub>) от  $n$  ( $n$  – длина массива) для метода Шелла, пирамидальной сортировки и метода Хоара (для массива случайных чисел) и сравнить построенные графики с графиком функции  $n \log n$ . \*.
5. Сравнить две версии метода Хоара по глубине рекурсии (при вычислении глубины рекурсии не путать ее с количеством рекурсивных вызовов) \*. Составить таблицу и проанализировать полученные результаты:

Таблица 9

$n$	Метод Хоара версия 1			Метод Хоара версия 2		
	Убыв.	Случ.	Возр.	Убыв.	Случ.	Возр.
100						
200						
300						
400						
500						

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 8

Тема: Метод прямого слияния

Цель работы: Изучение методов быстрой сортировки последовательностей.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить ручную сортировку методом прямого слияния. При тестировании программы можно использовать данный пример для проверки правильности реализации алгоритма.
2. Разработать подпрограмму сортировки последовательности целых чисел методом прямого слияния.
3. Проверить работу метода прямого слияния на последовательностях убывающих, возрастающих и случайных чисел и сделать вывод о зависимости или независимости метода от исходной упорядоченности

последовательности. Построить таблицу и проанализировать полученные результаты:

Таблица 10

Длина списка	(M <sub>ф</sub> +C <sub>ф</sub> ) метод прямого слияния		
	Возрастающие числа	Убывающие числа	Случайные числа
100			
200			
300			
400			
500			

4. Разработать следующие процедуры для работы со списком (очередью) \*:
  - заполнение списка (очереди) возрастающими числами;
  - заполнение списка (очереди) убывающими числами;
  - заполнение списка (очереди) случайными числами;
  - удаление всех элементов из списка.
6. Построить на экране в одной координатной плоскости графики зависимости трудоемкости (M<sub>ф</sub>+C<sub>ф</sub>) от  $n$  ( $n$  – длина массива или последовательности) для пирамидальной сортировки, метода Хоара и метода прямого слияния (для случайных чисел) ) и сравнить построенные графики с графиком функции  $n \log n$ . \*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 9

Тема: Метод цифровой сортировки

Цель работы: Изучение методов быстрой сортировки последовательностей.

1. Для набора из 12 трехзначных чисел в 6-ичной системе счисления выполнить вручную сортировку методом цифровой сортировки. При тестировании программы можно использовать данный пример для проверки правильности реализации алгоритма.
2. Разработать подпрограмму цифровой сортировки последовательности.
3. Применить цифровую сортировку для сортировки последовательности двухбайтовых и четырехбайтовых целых чисел в прямом и обратном порядке.
4. Применить цифровую сортировку для упорядочивания списка фамилий (строк)\*.
5. Экспериментально определить при каком количестве цифр в сортируемых числах цифровая сортировка начинает работать медленнее, чем метод Хоара.\*

## **ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 10**

Тема: Индексация и быстрый поиск

Цель работы: Изучение методов построения индексных массивов и быстрого поиска с использованием индексации.

1. Написать программу «Телефонный справочник», которая обрабатывает данные об абонентах телефонной станции. Каждый абонент имеет имя, адрес, телефонный номер. В программе описать массив абонентов (телефонный справочник). В справочнике должно быть не менее 10 элементов, которые заполняются либо программно, либо считываются из файла.
2. Разработать подпрограмму создания в памяти компьютера индексного массива для упорядочивания справочника (воспользоваться любой сортировкой, кроме пузырьковой). Применить разработанную подпрограмму для создания индексных массивов, упорядочивающих справочник по имени, адресу и номеру телефона абонента. Вывести на экран исходный массив абонентов и содержимое построенных индексных массивов.
3. Разработать подпрограмму вывода на экран упорядоченного справочника. Применить разработанную подпрограмму для вывода на экран справочника, упорядоченного по возрастанию имени абонента, адреса абонента и номера телефона абонента.
4. Разработать подпрограмму поиска в справочнике с использованием индексного массива. Применить разработанную подпрограмму для поиска абонента по имени, адресу и номеру телефона. Ключ для поиска вводить с клавиатуры.
5. Разработать подпрограмму добавления новой записи в справочник (данные вводить с клавиатуры).\*
6. Разработать подпрограмму удаления записи из справочника по номеру телефона абонента (номер вводить с клавиатуры).\*

## **ДЕРЕВЬЯ ПОИСКА**

### **ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 1**

Тема: Построение двоичного дерева. Вычисление характеристик дерева

Цель работы: Изучение процесса программного построения двоичного дерева.

1. Разместить в памяти компьютера двоичное дерево заданного вида (см. рисунок 1, номер задания соответствует номеру бригады), данные в вершинах заполнить случайными числами.



2. Написать подпрограммы для вычисления характеристик двоичного дерева, которые определяют
  - размер дерева;
  - высоту дерева;
  - среднюю высоту дерева;
  - контрольную сумму данных в вершинах дерева;
 и проверить их работу на конкретном примере.
3. Запрограммировать обход двоичного дерева слева направо и вывести на экран получившуюся последовательность данных.

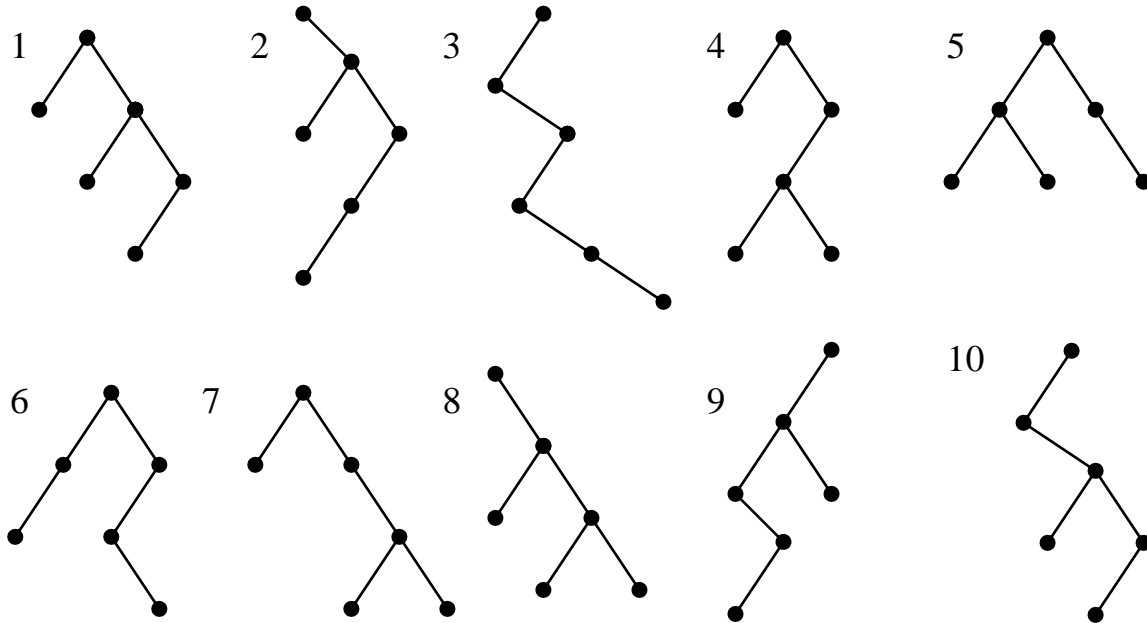


Рисунок 1 Варианты деревьев.

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 2

Тема: Идеально сбалансированное дерево поиска (ИСДП)

Цель работы: Изучение процесса программного построения ИСДП.

1. Для набора из 12 первых последовательных символов ФИО студента построить ИСДП. Для построенного дерева вычислить размер, контрольную сумму, высоту и среднюю высоту. В построенном дереве найти первую букву фамилии и букву «Я». При тестировании программы можно использовать данный пример для проверки правильности построения дерева.
2. Разработать подпрограмму построения идеально сбалансированного дерева поиска (ИСДП) для массива случайных чисел, а также логическую функцию для определения является ли данное двоичное дерево деревом поиска.

3. Разработать подпрограмму поиска вершины с заданным ключом в двоичном дереве поиска.
4. Построить ИСДП из 100, 200,..., 500 вершин. Распечатать обход дерева слева направо. Для построенных деревьев вычислить размер, контрольную сумму, высоту и среднюю высоту, используя разработанные функции. Заполнить таблицу и проанализировать полученные результаты:

Таблица 11

Размер дерева	ИСДП			
	Контр. сумма	Высота	Теор. оценка для средней высоты	Средняя высота
100				
200				
300				
400				
500				

5. Реализовать графическую подпрограмму изображения дерева на экране.\*

### ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 3

Тема: Случайное дерево поиска (СДП)

Цель работы: Изучение процесса программного построения СДП.

1. Построить СДП для набора из 12 первых последовательных символов ФИО студента. В построенном дереве найти первую букву фамилии и букву «Я». При тестировании программы можно использовать данный пример для проверки правильности построения дерева.
2. Разработать подпрограмму построения случайного дерева поиска (СДП) двумя способами: рекурсивно и с двойной косвенностью.
3. Построить СДП из 100, 200,..., 500 вершин (данные в вершинах произвольные, но все различные). Распечатать обход дерева слева направо.
4. Для построенного дерева вычислить размер, контрольную сумму, высоту и среднюю высоту, сравнить их с аналогичными характеристиками ИСДП. ИСДП необходимо строить для той же последовательности данных, что и СДП. Заполнить таблицу и проанализировать полученные результаты:

Таблица 12

Размер дерева	СДП			ИСДП		
	Контр.	Высота	Средняя	Контр.	Высота	Средняя

	сумма		высота	сумма		высота
100						
200						
300						
400						
500						

5. Применить подпрограмму графического изображения дерева на экране (из предыдущей лабораторной работы) для вывода случайного дерева поиска.\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 4

Тема: Случайное дерево поиска (СДП)

Цель работы: Изучение процесса программного удаления вершин СДП.

1. В построенном СДП для набора из 12 первых последовательных символов ФИО студента (см. предыдущую лабораторную работу) последовательно удалить все вершины в порядке их поступления в дерево. При тестировании программы можно использовать данный пример для проверки правильности построения дерева.
2. Разработать подпрограмму удаления из случайного дерева поиска (СДП) вершины с заданным ключом.
3. Реализовать удаление из СДП (100, 200,..., 500 вершин) всех вершин в порядке их поступления в дерево, распечатывать обход дерева слева направо после каждой удаленной вершины.
4. На базе СДП построить словарь частот встречаемости ключевых слов в тексте программы на Си.\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 5

Тема: Сбалансированные по высоте деревья поиска (АВЛ)

Цель работы: Изучение процесса программного построения АВЛ-дерева.

1. Построить АВЛ-дерево поиска для набора из 12 первых последовательных символов ФИО студента. В построенном дереве найти первую букву фамилии и букву «Я». При тестировании программы можно использовать данный пример для проверки правильности построения дерева.
2. Разработать подпрограмму построения АВЛ-дерева для массива целых чисел.

3. Построить AVL-дерево из 100, 200,..., 500 вершин (данные в вершинах произвольные, но все различные). Распечатать обход дерева слева направо.
4. Для построенного AVL-дерева вычислить размер, контрольную сумму, высоту и среднюю высоту, сравнить их с аналогичными характеристиками ИСП. ИСП необходимо строить для той же последовательности данных, что и AVL-дерево. Заполнить таблицу и проанализировать полученные результаты:

Таблица 13

Размер дерева	AVL-дерево			ИСП		
	Контр. сумма	Теор. оценки для сред. высоты	Средняя высота	Контр. сумма	Теор. оценки для сред. высоты	Средняя высота
100						
200						
300						
400						
500						

5. Применить подпрограмму графического изображения дерева на экране (из предыдущей лабораторной работы) для вывода AVL-дерева.\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 6

Тема: Сбалансированные по высоте деревья поиска (AVL)

Цель работы: Изучение процесса программного удаления вершин AVL-дерева.

1. В построенном AVL-дереве для набора из 12 первых последовательных символов ФИО студента (см. предыдущую лабораторную работу) последовательно удалить все вершины в порядке их поступления в дерево. При тестировании программы можно использовать данный пример для проверки правильности удаления из AVL-дерева.
2. Разработать подпрограмму удаления из AVL-дерева вершины с заданным ключом.
3. Реализовать удаление из AVL-дерева (100, 200,..., 500 вершин) всех вершин в порядке их поступления в дерево, распечатывать обход дерева слева направо после каждой удаленной вершины.
4. Подтвердить экспериментально утверждение, что при добавлении вершин в AVL-дерево на каждые два включения встречается один поворот, а при

удалении вершин поворот происходит в одном случае из пяти. Заполнить таблицу и проанализировать полученные данные: \*

Таблица 14

Количество добавленных вершин	Количество поворотов при добавлении	Частота поворотов при добавлении	Количество удаленных вершин	Количество поворотов при удалении	Частота поворотов при удалении
100			100		
200			200		
300			300		
400			400		
500			500		

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 7

Тема: Двоичное Б-дерево поиска (ДБД)

Цель работы: Изучение процесса программного построения ДБД.

1. Построить ДБД для набора из 12 первых последовательных символов ФИО студента. В построенном дереве найти первую букву фамилии и букву «Я». При тестировании программы можно использовать данный пример для проверки правильности построения дерева.
2. Разработать подпрограмму построения ДБ-дерева для массива целых чисел
3. Построить ДБ-дерево из 100, 200,..., 500 вершин (данные в вершинах произвольные, но все различные). Распечатать обход дерева слева направо.
4. Для построенного ДБ-дерева вычислить размер, контрольную сумму, высоту и среднюю высоту (как для двоичного дерева) и высоту ДБ-дерева как количество уровней, сравнить их с аналогичными характеристиками АВЛ-дерева. ДБ-дерево необходимо строить для той же последовательности данных, что и АВЛ-дерево. Заполнить таблицу и проанализировать полученные результаты:

Таблица 15

Размер дерева	АВЛ-дерево		ДБД			
	Контр. сумма	Сред. высота	Контр. сумма	Кол-во уровней	Теор. оценки для высоты ДБД	Сред. высота для дв. дерева.
100						
200						
300						

400						
500						

5. Реализовать графическую подпрограмму изображения ДБ-дерева на экране.\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 8

Тема: Б-дерево поиска произвольного порядка

Цель работы: Изучение процесса программного построения Б-дерева.

1. Для набора из 12 первых последовательных символов ФИО студента выполнить вручную построение Б-дерева порядка  $m=2$ . В построенном дереве найти первую букву фамилии и букву «Я». При тестировании программы использовать данный пример для проверки правильности реализации алгоритма.
2. Разработать подпрограмму построения Б-дерева порядка  $m$  ( $m \geq 2$ ). Страницу Б-дерева реализовать в виде массива.
3. Построить Б-дерево из 100 вершин для различных значений  $m$ , вывести на экран результаты обхода Б-дерева слева направо.
4. Для Б-дерева порядка  $m$  реализовать подпрограммы вычисления размера (количество вершин дерева), контрольной суммы (сумма всех элементов дерева), высоты (количество уровней дерева) и средней высоты (среднее количество операций сравнения при поиске). Заполнить таблицу и проанализировать полученные результаты.

Таблица 16

Порядок Б-дерева	Размер	Контр. Сумма	Высота	Средняя высота
2	100			
3	100			
4	100			
5	100			

5. Реализовать подпрограмму графического изображения Б-дерева порядка  $m$  ( $m \geq 2$ ) на экране.\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 9

Тема: Дерево оптимального поиска (ДОП)

Цель работы: Изучение процесса программного построения ДОП (точный алгоритм).

1. Построить ДОП точным алгоритмом для набора из всех символов ФИО студента (вес символа – количество этого символа в наборе), вычислить средневзвешенную высоту построенного дерева. В построенном дереве найти первую букву фамилии и букву «Я». При тестировании программы можно использовать данный пример для проверки правильности построения дерева.
2. Разработать подпрограмму построения дерева оптимального поиска точным алгоритмом.
3. Построить дерево оптимального поиска из 100, 200,..., 500 вершин (данные в вершинах произвольные, но все различные), распечатать обход дерева слева направо.
4. Для построенного дерева вычислить размер, контрольную сумму и средневзвешенную высоту. Заполнить таблицу и проанализировать полученные результаты:

Таблица 17

Размер дерева	ДОП	
	Контрольная Сумма	Средневзвешенная Высота
100		
200		
300		
400		
500		

5. Реализовать графическую подпрограмму изображения дерева оптимального поиска на экране (с весом каждой вершины).\*

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ 10

Тема: Дерево оптимального поиска (приближенные алгоритмы)

Цель работы: Изучение процесса программного построения почти оптимальных деревьев поиска.

1. Построить ДОП двумя приближенными алгоритмами из всех символов ФИО студента (вес символа – количество этого символа в наборе), вычислить средневзвешенные высоты построенных деревьев. В построенных деревьях найти первую букву фамилии и букву «Я». При тестировании программы можно использовать данный пример для проверки правильности построения дерева.
2. Реализовать программно алгоритмы A1 и A2 для построения почти оптимальных деревьев поиска.
3. Построить почти оптимальные деревья поиска из 100, 200,..., 500 вершин (данные в вершинах произвольные, но все различные) с помощью алгоритмов A1 и A2, распечатать их обход слева направо.

4. Для построенных деревьев вычислить размер, контрольную сумму и средневзвешенную высоту, сравнить их с аналогичными характеристиками дерева оптимального поиска. Заполнить таблицу и проанализировать полученные результаты:

Таблица 18

Размер дерева	A1		A2	
	Контр. сумма	Средне- взвешенная высота	Контр. сумма	Средне- взвешенная высота
100				
200				
300				
400				
500				

5. Применить графическую подпрограмму изображения дерева на экране для почти оптимальных деревьев.\*