**Лабораторная работа 5.** **Построение 2-3-4 дерева.**

**Цель работы: Закрепление теоретических знаний и получение практических навыков создания программ для построения 2-3-4 деревьев.**

**Краткие теоретические сведения**

В двоичном дереве каждый узел содержит один элемент данных и может иметь до двух потомков. Дерево с большим количеством элементов данных и потомков называется *многопутевым* деревом. *Деревья 2-3-4* представляют собой многопутевые деревья, у которых каждый узел может иметь до четырех потомков и трех элементов данных.

Деревья 2-3-4 представляют интерес по нескольким причинам.

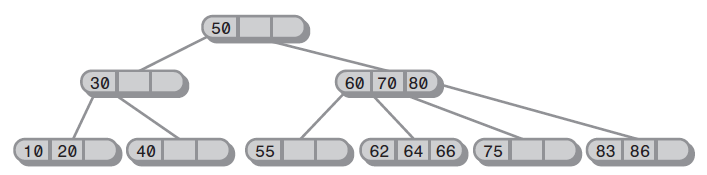
Во-первых, это сбалансированные деревья, которые ведут себя как красно-черные деревья. Они обладают чуть меньшей эффективностью, чем красно-черные деревья, но проще программируются.

Во-вторых (что самое важное), они упрощают изучение B-деревьев.

B-дерево представляет собой другую разновидность многопутевых деревьев, особенно удобную для организации данных во внешней памяти (то есть за пределами оперативной памяти; обычно под внешней памятью понимается дисковое устройство). Узел B-дерева может иметь десятки и даже сотни потомков.

1. **Знакомство с деревьями 2-3-4**

На рис. 1 изображено небольшое дерево 2-3-4. Каждый узел (в форме прямоугольника с закругленными краями) может содержать один, два или три элемента данных.



**Рис. 1.** Дерево 2-3-4

У трех верхних узлов имеются потомки, а все шесть узлов из нижнего ряда являются листовыми, то есть по определению не имеют потомков. В дереве 2-3-4 все листовые узлы всегда находятся на одном уровне.

Цифры 2, 3 и 4 в названии дерева обозначают количество связей с потомками, которые могут содержаться в заданном узле. Для не-листовых узлов возможны три конфигурации:

* Узел с одним элементом данных всегда имеет двух потомков.
* Узел с двумя элементами данных всегда имеет трех потомков.
* Узел с тремя элементами данных всегда имеет четырех потомков.

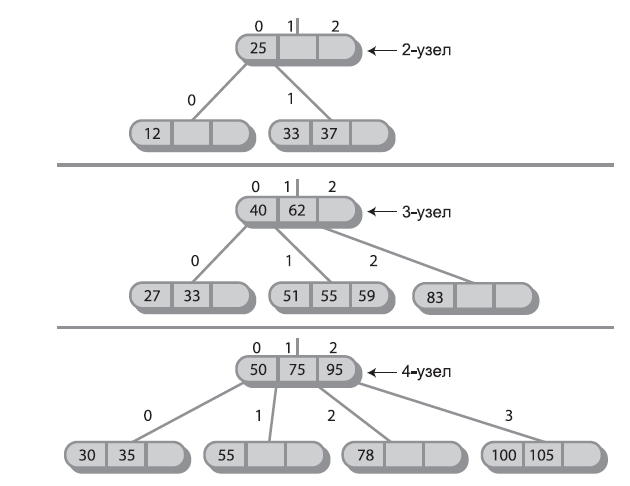
Короче говоря, количество потомков у не-листового узла всегда на единицу больше количества элементов данных. Или если выразить это соотношение в виде формулы, соотношение количества потомков L с количеством элементов данных D выглядит так:

L = D + 1.

Это важнейшее соотношение определяет структуру деревьев 2-3-4. С другой стороны, листовые деревья не имеют потомков, но могут содержать один, два или три элемента данных. Пустые узлы запрещены.

Поскольку узлы дерева 2-3-4 могут содержать до четырех потомков, такое дерево называется *многопутевым деревом порядка 4*.

Между двоичными деревьями и деревьями 2-3-4 существует важное различие (помимо максимального количества узлов): в двоичном дереве узел может иметь до двух потомков. Одна связь с левым или правым потомком абсолютно допустима; другая ссылка при этом равна null.



**Рис. 2.** Узлы дерева 2-3-4

В деревьях 2-3-4 узлы с одной связью невозможны. Узел с одним элементом данных всегда должен иметь две связи (если только узел не является листовым — в этом случае он вообще не имеет связей).

Возможные варианты представлены на рис. 2. Узел с двумя связями называется 2-узлом, узел с тремя связями называется 3-узлом, а узел с четырьмя связями называется 4-узлом, но такого понятия, как 1-узел, не существует.

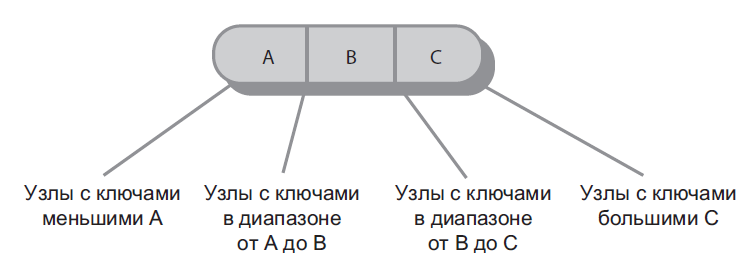
**Структура дерева 2-3-4**

Для удобства мы пронумеруем элементы данных в узле от 0 до 2, а ссылки на потомков — от 0 до 3, как показано на рис. 2. Элементы данных узла упорядочиваются по возрастанию ключа; обычно значения располагаются слева направо.

Важным аспектом любой древовидной структуры является связь ее ссылок с ключами элементов данных. В двоичном дереве все потомки с ключом, меньшим ключа узла, содержатся в поддереве, корнем которого является левый потомок узла; соответственно все потомки с ключом, большим ключа узла, содержатся в поддереве, корнем которого является правый потомок. В дереве 2-3-4 действует тот же принцип, но в немного расширенном виде:

* У всех потомков поддерева, корнем которого является узел 0, ключи меньше ключа 0.
* У всех потомков поддерева, корнем которого является узел 1, ключи больше ключа 0, но меньше ключа 1.
* У всех потомков поддерева, корнем которого является узел 2, ключи больше ключа 1, но меньше ключа 2.
* У всех потомков поддерева, корнем которого является узел 3, ключи больше ключа 2.

Отношения наглядно представлены на рис. 3. Дубликаты в деревьях 2-3-4 обычно запрещены, поэтому беспокоиться о равенстве ключей при сравнении не нужно.



**Рис. 3.** Ключи и потомки

Вернемся к дереву на рис. 1. Как и во всех деревьях 2-3-4, листья в нем располагаются на одном уровне (нижний ряд). Узлы верхних уровней часто бывают неполными, то есть содержат только один или два элемента данных вместо трех.

Также заметьте, что дерево сбалансировано. Баланс сохраняется даже при вставке данных, упорядоченных по возрастанию (или убыванию). Самобалансируемость дерева 2-3-4 обусловлена новым способом вставки данных.

* 1. **Основные операции в дереве 2-3-4**

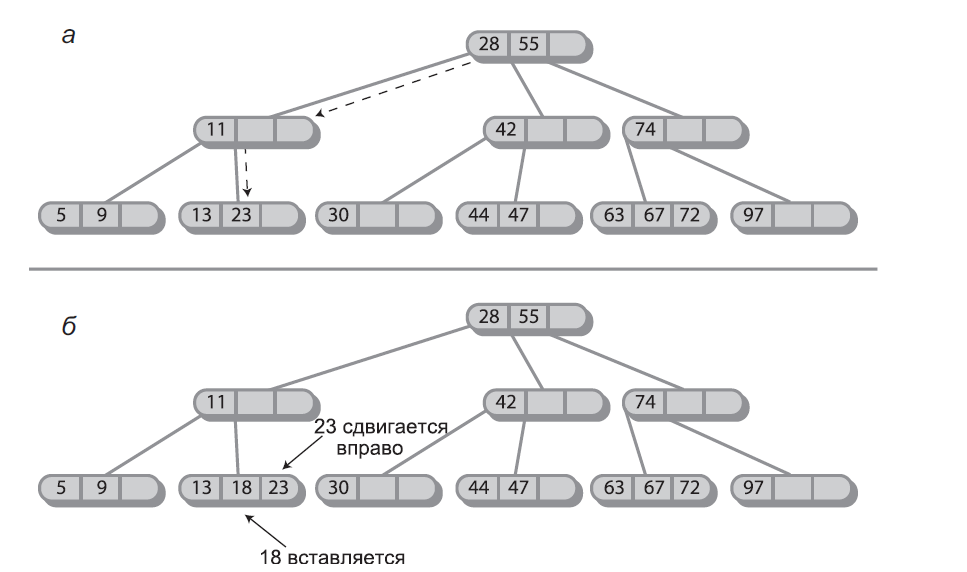
**Поиск в дереве 2-3-4**

Поиск элемента данных с заданным ключом напоминает поиск в двоичном дереве. Алгоритм начинает с корня дерева и вплоть до обнаружения ключа выбирает ссылку, ведущую к поддереву с соответствующим диапазоном значений. Например, чтобы провести поиск элемента данных с ключом 64 в дереве на рис. 1, следует начать с корня. При сравнении с корневым ключом элемент не находится. Так как 64 больше 50, алгоритм переходит к потомку 1, которого мы обозначим 60/70/80. (Стоит напомнить, что потомок 1 является правым, потому что нумерация потомков и связей начинается слева.) В этом узле искомый элемент также отсутствует, поэтому необходимо перейти к следующему потомку. На этот раз 64 больше 60, но меньше 70, поэтому алгоритм снова переходит к потомку 1. На этот раз искомый элемент находится в узле 62/64/66.

**Вставка**

Новые элементы данных всегда вставляются в листьях, находящиеся в нижнем ряду дерева. Если узлы вставляются в узлах с потомками, то количество потомков необходимо изменить для сохранения структуры дерева — согласно правилу, гласящему, что количество потомков должно быть на единицу больше количества элементов данных в узле.

В одних случаях вставка в дереве 2-3-4 выполняется легко, в других она основательно усложняется. Процесс всегда начинается с поиска соответствующего листового узла. Если в процессе поиска полные узлы не обнаружены, то вставка выполняется легко — при достижении подходящего листового узла новый элемент данных просто вставляется в него. На рис. 4 элемент данных с ключом 18 вставляется в дерево 2-3-4.

**

**Рис. 4.** Вставка без разбиений: *а* — до вставки; *б* — после вставки

Вставка может потребовать перемещения одного или двух других элементов данных в узлах, чтобы после вставки нового элемента ключи следовали в правильном порядке. В приведенном примере элемент 23 сдвигается вправо, чтобы освободить место для 18.

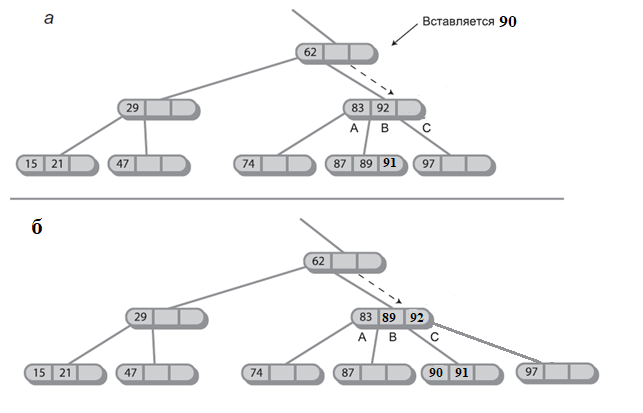
**Разбиение узлов**

Если на пути вниз к позиции вставки встречается заполненный узел, ситуация усложняется. Такие узлы должны *разбиваться*; именно процесс разбиения сохраняет сбалансированность дерева. Разновидность деревьев 2-3-4, часто называется *нисходящими деревьями 2-3-4*, потому что узлы разбиваются в процессе перемещения вниз к позиции вставки.

Назовем элементы данных в разбиваемом узле A, B и C. Процесс разбиения происходит следующим образом (предполагается, что разбиваемый узел не является корневым; разбиение корневого узла будет рассмотрено позднее).

* Создается новый пустой узел. Он является «братом» (одноуровневым узлом) по отношению к разбиваемому узлу и размещается справа от него.
* Элемент данных C перемещается в новый узел.
* Элемент данных B перемещается в родителя разбиваемого узла.
* Элемент данных A остается на своем месте.
* Два правых потомка отсоединяются от разбиваемого узла и связываются с новым узлом.

Пример разбиения представлен на рис. 5. Разбиение также можно описать как преобразование 4-узла в два 2-узла.

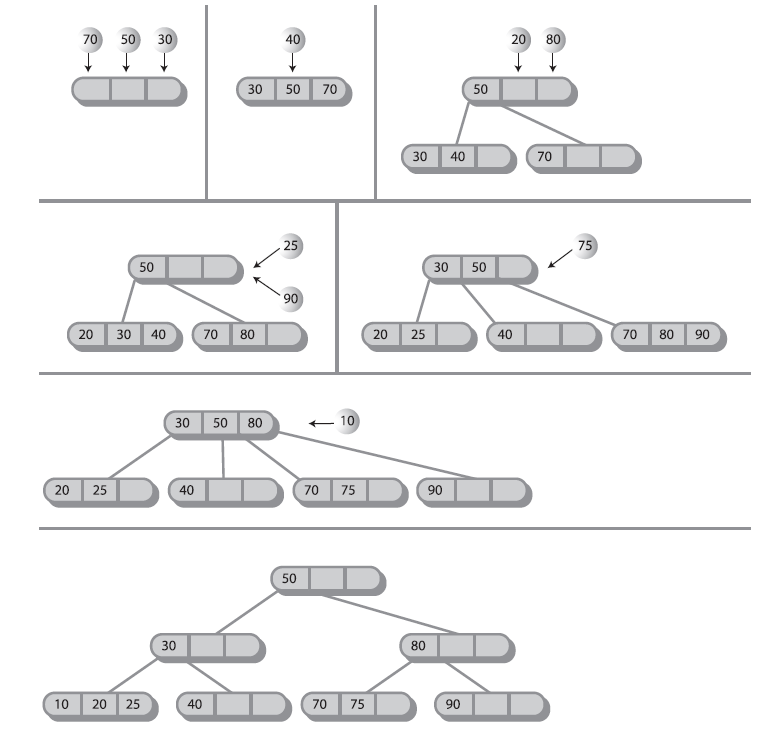
**

**Рис. 5.** Разбиение узла: *а* — до вставки; *б* — после вставки

Обратите внимание: в результате разбиения узла данные смещаются направо и вверх. Такое перемещение обеспечивает сбалансированность дерева.

В этом примере вставка требует разбиения только одного узла, но на пути к позиции вставки может быть обнаружено несколько полных узлов. В таких ситуациях приходится выполнять несколько разбиений.

На рис. 6 показана серия вставок в пустое дерево. Всего выполняются четыре разбиения: два для корневого узла и два для листьев.



**Рис. 6.** Вставка в дерево 2-3-4

**Пример построения 2-3-4 дерева**

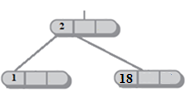
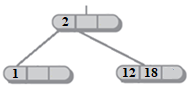
Дано последовательность чисел 1, 2, 18, 13, 12, 21, 22, 33, 15, 19, 32, 37, 88, 11, 4, 25, 38, 29, 81, 31

Требуется построить 2-3-4 дерево, внося числа последовательно слева направо

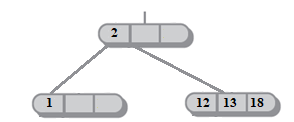
Шаги 1, 2, 3 Дерево пустое, поэтому последовательно вносим числа 1, 2, 18 в корневой узел.



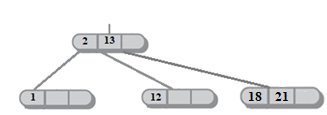
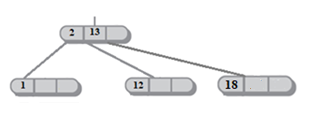
Шаг 4. Вставка числа 13. Корневой узел заполнен, поэтому он разделяется, а затем в правый лист вставляется число 13.

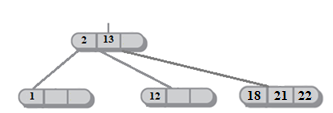
Шаг 5. Вставка числа 12. Число вставляется в правый лист.



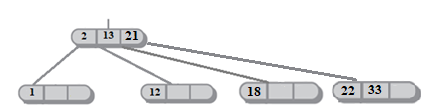
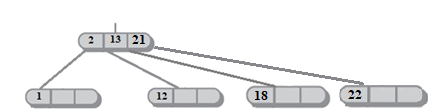
Шаг 6. Вставка числа 21. При движении вниз встретился полностью заполненный лист – он разбивается, а затем в правый лист вставляется число 21.



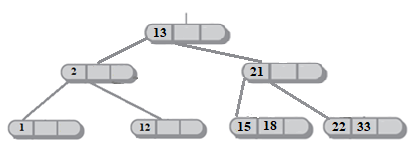
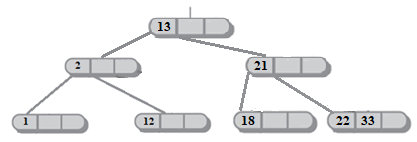
Шаг 7. Вставка числа 22. Число вносится в третий лист



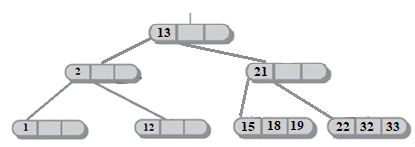
Шаг 8. Вставка числа 33. При движении вниз встретился полностью заполненный лист – он разбивается и в четвертый лист вставляется число 33.



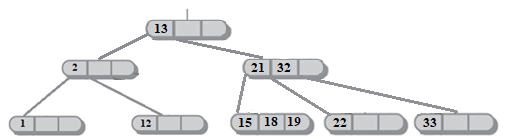
Шаг 9. Вставка числа 15. При движении вниз встретился полностью заполненный корневой узел. Он разбивается, затем число 15 вставляется в третий лист.

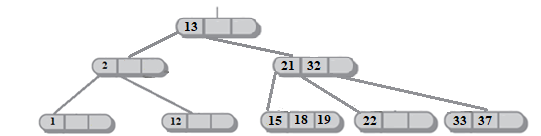


Шаги 10 и 11. Вставка чисел 19 и 32. Число 19 вставляется в третий лист, а число 32 в четвертый лист.

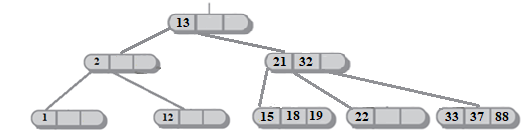


Шаг 12. Вставка числа 37. При движении вниз встречается полностью заполненный лист (четвертый), он разбивается, а затем число 37 вставляется в пятый лист

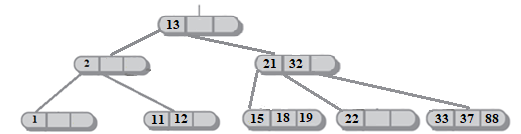




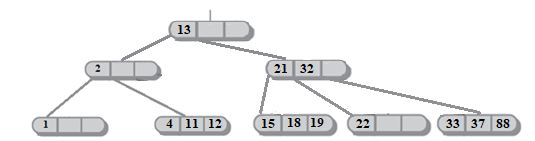
Шаг 13. Вставка числа 88. Число 88 вставляется в пятый лист.



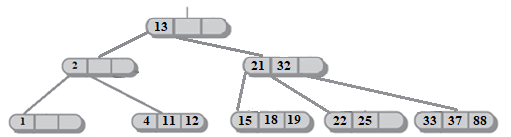
Шаг 14. Вставка числа 11. Число 11 вставляется во второй лист.



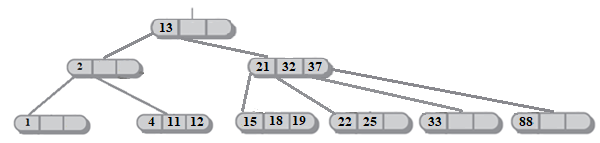
Шаг 15. Вставка числа 4. Число 4 вставляется во второй лист.

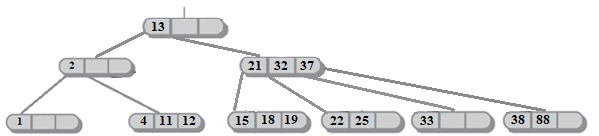


Шаг 16. Вставка числа 25. Число 25 вставляется в четвертый лист.

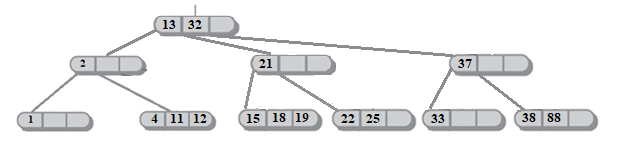


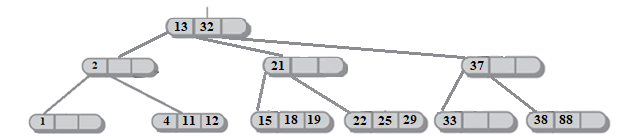
Шаг 17. Вставка числа 38. При движении вниз встречается полностью заполненный лист (пятый), он разбивается, а затем число 38 вставляется в шестой лист



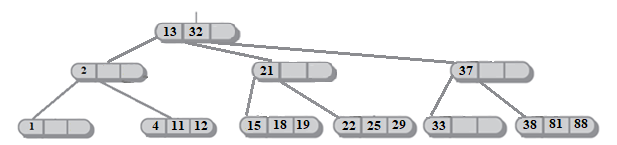


Шаг 18. Вставка числа 29. При движении вниз встречается полностью заполненный узел (правый потомок корня), он разбивается, а затем число 29 вставляется в четвертый лист

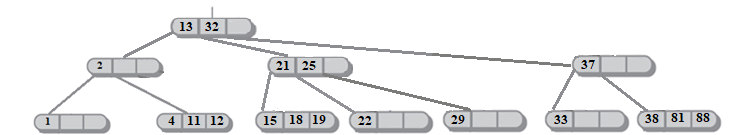


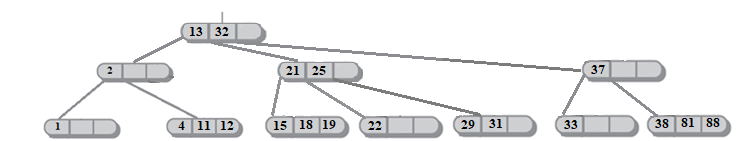


Шаг 19. Вставка числа 81. Число вставляется в шестой лист



Шаг 20. Вставка числа 31. При движении вниз встречается полностью заполненный лист (четвертый), он разбивается, а затем число 31 вставляется в пятый лист





B-дерево построено. Высота дерева 2.

**2. Задание на лабораторную работу.**

* + 1. Построить 2-3-4 дерево из последовательности целых чисел (по варианту), внося их в дерево последовательно слева направо и изобразить это дерево.
    2. Изобразить 2-3-4 дерево после внесения каждого числа.

Варианты задания

|  |  |
| --- | --- |
| Вариант | Последовательность чисел |
| 1 | 4, 6, 2, 8, 11, 15, 12, 21,17, 13, 35, 37, 45, 87, 92, 29, 77, 76, 75, 79 |
| 2 | 10, 12, 8, 3, 2, 21, 22, 43, 15, 19, 1, 15 18, 25, 27, 28, 31, 33, 35, 36 |
| 3 | 22, 25, 28, 15, 13, 5, 2, 11, 17, 23, 5, 7,  15, 33, 34, 35, 36, 45, 54, 55 |
| 4 | 12, 3, 15, 17, 1, 23, 43, 7, 5, 11, 19, 24, 26, 31, 32, 34, 54, 55, 58, 60 |
| 5 | 18, 16, 28, 12, 14, 33, 41, 25, 8, 2, 4, 6, 7, 8, 21, 23, 31, 35, 46, 47 |
| 6 | 32, 50, 55, 30, 51, 67, 72, 94, 81, 35, 1, 3, 7, 8, 9, 15, 11, 12, 23, 24 |
| 7 | 57, 59, 35, 30, 70, 25, 28, 45, 23, 89, 6, 5, 3, 15, 17, 19, 34, 36, 37, 41 |
| 8 | 78, 55, 92, 96, 30, 98, 97, 43, 54, 8, 9, 4  11, 18, 12, 13, 38, 91, 47, 39 |
| 9 | 12, 20, 6, 24, 30, 35, 40, 7, 4, 23, 51, 52, 53, 98, 9, 10, 21, 28, 33, 99 |
| 10 | 40, 51, 35, 30, 27, 22, 54, 63, 76, 3, 4, 6,  19, 18, 16, 65, 68, 75, 77, 99 |
| 11 | 33, 36, 41, 27, 25, 13, 7, 15, 9, 55, 6, 5,  9, 19, 20, 47, 39, 49, 55, 56 |
| 12 | 72, 16, 77, 85, 12, 95, 91, 63, 19, 47 |
| 13 | 77, 81, 53, 45, 33, 39, 99, 22, 28, 9, 8, 7, 34, 35, 21, 26, 29, 85, 88, 95 |
| 14 | 54, 64, 70, 43, 77, 12, 15, 7, 29, 33, 8, 9,  13, 15, 17, 14, 21, 22, 35, 76 |
| 15 | 49, 47, 54, 45, 46, 32, 24, 12, 19, 7, 2, 3,  11, 19, 18, 25, 27, 28, 74, 75 |

**3. Содержание отчета по лабораторной работе**

- наименование лабораторной работы и ее цель;

- задание на лабораторную работу согласно варианту;

- изображения дерева после добавления каждого числа;

- выводы.

1. Контрольные вопросы
2. Какое дерево называется 2-3-4 деревом?
3. Сколько элементов может содержать узел 2-3-4 дерева?
4. Сколько потомков может иметь узел 2-3-4 дерева?
5. Как связаны ключи узла с потомками в 2-3-4 дереве?
6. Что такое разбиение узлов и как оно производится в 2-3-4 дереве?
7. Какова сложность поиска в B-дереве?