**Методичні рекомендації**

**до виконання лабораторної роботи №№ 5-6**

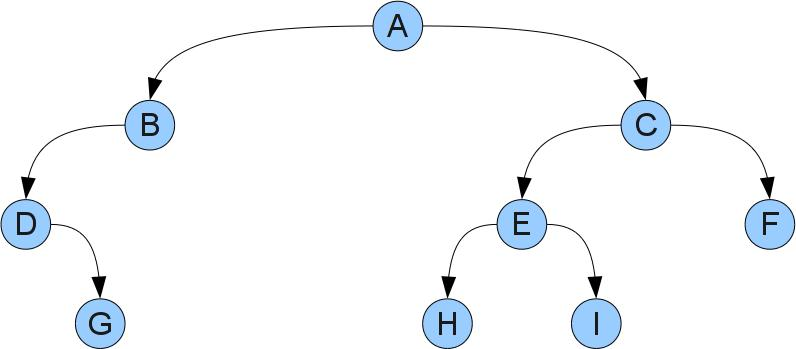
**Тема:** Складання та налагодження програми рішення задачі створення та обходу бінарних дерев.

**Мета:** Отримання навиків в організації динамічної структури даних – бінарного дерева.

ТЕОРЕТИЧНІ ВІДОМОСТІ

***Бінарне дерево*** – це структура даних, кожен елемент якої окрім самих даних містить покажчики на два наступних елементи структури. Один з цих наступних елементів умовно називається лівим, а інший - правим.

Конкретніше, у дерева є виділена вершину-корінь й у кожної вершини може бути лівий і правого синів. На словах звучить трохи складно, але якщо глянути на картинку все стає зрозумілим:



У цього дерева коренем буде вершина A. Видно, що у вершини D відсутній лівий син, у вершини B - правий, а у вершин G, H, F й Й - обоє. Вершини без синів прийнято називати листами.

Кожній вершині X можна зіставити своє дерево, що складається з вершини, її синів, синів її синів, і т.д. Таке дерево називають піддеревом з коренем X. Лівим і правим піддеревами X називають піддерева з коріннями відповідно в левом і правом синах X. Помітимо, що такі піддерева можуть виявитися порожніми, якщо в X немає відповідного сина.

Дані в дереві зберігаються в його вершинах. У програмах вершини дерева звичайно представляють структурою, що зберігає дані й два посилання на лівого й правого сина. Відсутні вершини позначають null.

Множина всіх вузлів, рівновіддалених від кореня, називається рівнем. Вузол, з якого не починається жодна вітка, називається кінцевим або листовим вузлом.

Оскільки дерево бінарне, кожен вузол може породжувати два вузли наступного рівня. Породжені вузли є дочірніми по відношенню до вузла, що їх породив. Породжуючий вузол є батьківським по відношенню до своїх дочірніх вузлів. Батьківський вузол разом із своїми дочірніми складає ланку.

Сумарна кількість рівнів дерева називається висотою дерева.

Основними операціями при роботі з деревами є:

-додавання елемента до дерева;

-пошук елемента дерева, що відповідає заданому критерію пошуку;

-сортування елементів дерева;

-вилучення елемента дерева.

Процес доступу до елементів дерева називається проходженням дерева. Існує три способи проходження дерев:

* обхід у прямому порядку,
* обхід у зворотному порядку,
* кінцевий (симетричний) обхід.

***Прямий порядок*** проходження бінарного дерева ("униз", "у глибину"):



1. потрапити в корінь,

2. пройти в прямому порядку ліве піддерево,

3. пройти в прямому порядку праве піддерево.

Проходження бінарного дерева у ***зворотному порядку*** ("нагору"):



1. пройти у зворотному порядку ліве піддерево

2. потрапити в корінь

3. пройти у зворотному порядку праве піддерево.

***Кінцевий (симетричний)*** обхід дерева



1. пройти в симетричному порядку ліве піддерево,

2. пройти в симетричному порядку праве піддерево,

3. потрапити в корінь.

Видалення бінарного дерева.

Слід враховувати, що при видаленні вузла дерева спочатку повинні бути видалені його лівий та правий нащадок, а потім видаляється сам вузол Отже для реалізації цієї операції слід використовувати кінцевий (симетричний) обхід дерева. Рекурсивна функція видалення бінарного дерева наведена в лістингу 5.

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1. Провести аналіз поставленої задачі.

Загальна постановка завдання:

Скласти та налагодити програму створення та обробки бінарного дерева на основі нелінійного списку за алгоритмом згідно Вашого варіанту. Програма повинна задовольняти наступним вимогам:

* організувати користувацьке меню, яке повинно містити наступні пункти:

1. Створення бінарного дерева.
2. Перегляд вмісту бінарного дерева.
3. Обробка дерева згідно Вашого варіанту.

* забезпечити коректне введення користувачем вхідних даних;

− при обробці дерева враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;

− введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

1. Розробити та налагодити програму рішення задачі.
2. Оформити звіт з лабораторної роботи.

**Звіт повинен містити наступні розділи:**

1. Постановка задачі.
2. Текст програми з відповідними коментарями.
3. Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють всі можливі варіанти роботи програми та помістити в звіт копії вікон виконання програми для кожного тестового набору.

1. Висновок.

**Варіанти завдань:**

1. Створити бінарне дерево натуральних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, підрахувати кількість парних чисел в дереві та вивести ці числа на екран.
2. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, від’ємні числа дерева заміни їх квадратами. Вивести оновлене дерево на екран.
3. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи симетричний обхід дерева, знайти максимальний елемент дерева та вивести його на екран.
4. Створити бінарне дерево натуральних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, підрахувати кількість непарних чисел в дереві та вивести ці числа на екран.
5. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, знайти максимальний та мінімальний елементи дерева та вивести ці значення на екран.
6. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи симетричний обхід дерева, знайти найбільше від’ємне число дерева та вивести його на екран.
7. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, додатні числа дерева піднести до третього ступеня. Вивести оновлене дерево на екран.
8. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, підрахувати кількість листових вузлів дерева.
9. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Вивести створене дерево на екран. Використовуючи симетричний обхід дерева, кожному листовому вузлу дерева додати лівого прямого нащадка зі значенням, вказаним користувачем. Вивести оновлене дерево на екран.
10. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, підрахувати кількість вузлів дерева, які мають тільки правого прямого нащадка.
11. Створити бінарне дерево натуральних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, підрахувати суму парних чисел в дереві.
12. Створити бінарне дерево натуральних чисел введенням даних з клавіатури. Використовуючи симетричний обхід дерева, підрахувати суму парних чисел в дереві. Значення кореневого вузла дерева замінити на значення отриманої суми.
13. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, знайти максимальний елемент дерева та вивести його на екран. Всі додатні числа дерева заміни максимальним елементом. Вивести оновлене дерево на екран.
14. Створити бінарне дерево натуральних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, підрахувати добуток непарних чисел в дереві. Замінити значення кореневого вузла цим добутком. Вивести оновлене дерево на екран.
15. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи симетричний обхід дерева, знайти суму максимального та мінімального елементів дерева та вивести це значення на екран.
16. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, підрахувати кількість вузлів дерева, в яких правий або лівий прямий потомок є листовим вузлом.
17. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Вивести створене дерево на екран. Кожному листовому вузлу дерева додати правого прямого нащадка зі значенням, вказаним користувачем. Вивести оновлене дерево на екран.
18. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи симетричний обхід дерева, підрахувати кількість вузлів дерева, які мають тільки лівого прямого нащадка.
19. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, знайти максимальний та мінімальний елементи дерева та вивести ці значення на екран.
20. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, додатні числа дерева помножити на -1. Вивести оновлене дерево на екран.
21. Створити бінарне дерево натуральних чисел введенням даних з клавіатури. Використовуючи симетричний обхід дерева, підрахувати кількість непарних чисел в дереві та вивести ці числа на екран.
22. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, знайти мінімальний елемент дерева та вивести його на екран. Всі від’ємні числа дерева заміни мінімальним елементом. Вивести оновлене дерево на екран.
23. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, підрахувати кількість вузлів дерева, які мають як лівого так і правого прямих потомків.
24. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи симетричний обхід дерева, знайти мінімальний елемент дерева та вивести його на екран. Значення всіх листових вузлів дерева замінити мінімальним елементом. Вивести оновлене дерево на екран.
25. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Вивести на екран значення лівого та парового прямих потомків кореневого вузла дерева.
26. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, значення листових вузлів дерева заміни їх квадратами. Вивести оновлене дерево на екран.
27. Створити бінарне дерево натуральних чисел введенням даних з клавіатури. Використовуючи симетричний обхід дерева, підрахувати суму парних чисел в дереві. Значення кореневого вузла дерева замінити на значення отриманої суми.
28. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи прямий обхід дерева, знайти максимальний елемент дерева. Підрахувати кількість вузлів дерева, значення яких дорівнює максимальному.
29. Створити бінарне дерево дійсних чисел введенням даних з клавіатури. Використовуючи зворотній обхід дерева, значення листових вузлів дерева помножити на -1. Вивести оновлене дерево на екран.

КОНТРОЛЬНІ ПИТАННЯ

1 Яку структуру мають вершини двійкового дерева?

2 Чому для дерев існує кілька правил обходу вершин?

3 Які правила обходу вершин дерева є основними?

4 Як виконується обхід дерева в прямому напрямку?

5 Як виконується обхід дерева в симетричному напрямку?

6 Як виконується обхід дерева у зворотному напрямку?

7 Як виконується обхід дерева в назад-симетричному напрямку?

8 Чому рекурсивна реалізація правил обходу є найбільш зручної?

9 Що відбувається при рекурсивному виконанні обходу дерева?

10 Як програмно реалізується обхід дерева в прямому напрямку?

11 Як програмно реалізується обхід дерева в симетричному напрямку?

12 Як програмно реалізується обхід дерева у зворотному напрямку?

13 Який формальний параметр необхідний для рекурсивної реалізації правил обходу і як він використається?

14 Як правильно виконати знищення всієї деревоподібної структури?

Розробив: Старосельцева О.В.

Розглянуто та схвалено

на засіданні предметної (циклової) комісії

комп’ютерної інженерії

Протокол № 1 від 30.08.2017 р.

Голова комісії \_\_\_\_\_\_\_\_ Старосельцева О.В.