

# Дисципліна: “Програмування складних алгоритмів”

---

**Лабораторна робота №7. Поняття бінарного дерева.  
Обхід бінарного дерева. Створення, відображення дерева.  
Вставлення, видалення елементів у бінарному дереві.**

**Мета роботи:** набути навичок створення та обробки бінарних дерев.

**Завдання до лабораторної роботи:**

Розробити спосіб економного зберігання в пам'яті розріджених матриць.

Виконати індивідуальне завдання над стисненою матрицею.

Вивести матрицю до та після обробки у стисненому та розгорнутому вигляді.



# Дисципліна: “Програмування складних алгоритмів”

---

## Послідовність виконання роботи:

1. Ознайомитись із теоретичними відомостями.
2. Виконати програмування програм за поданими прикладами.  
(Застосування набутих знань).
3. Виконати варіант самостійної роботи. (Закріплення набутих знань).
4. Оформити звіт на виконану роботу. (Узагальнення та систематизація набутих знань).



# Теоретичні відомості

**Дерево** – це структура даних, що являє собою сукупність елементів і відносин, що утворюють ієрархічну структуру цих елементів

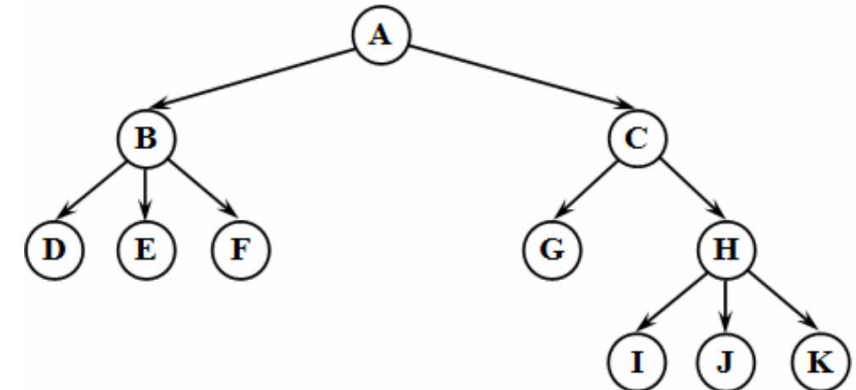
Кожен елемент дерева називається *вершиною* (*вузлом*) дерева. Вершини дерева з'єднані спрямованими дугами, які називають гілками дерева.

Початковий вузол дерева називають *коренем* дерева, йому відповідає нульовий рівень. Листями дерева називають вершини, в які входить одна гілка і не виходить жодної гілки.

Кожне дерево має такі властивості:

- 1) існує вузол, в який не входить ні одна дуга (корінь);
- 2) у кожен вузол, крім кореня, входить одна дуга.

Дерева особливо часто використовують на практиці при зображенні різних ієрархій.

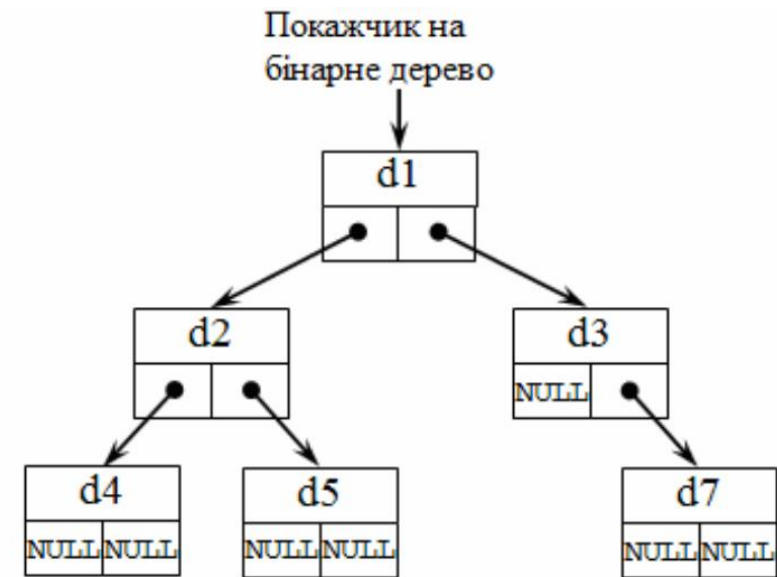
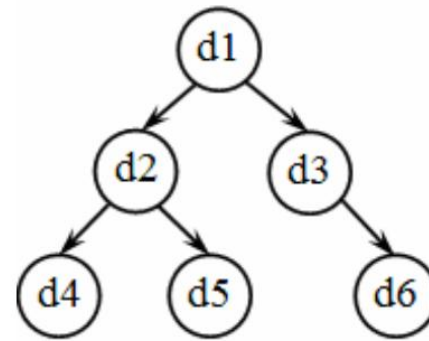


# Теоретичні відомості

**Бінарні дерева** є деревами зі ступенем не більше двох.

**Бінарне (двійкове)** дерево – це динамічна структура даних, що являє собою дерево, в якому кожна вершина має не більше двох нащадків.

Таким чином, бінарне дерево складається з елементів, кожен з яких містить інформаційне поле і не більше двох посилань на різні бінарні піддерева. На кожен елемент дерева є рівно одне посилання





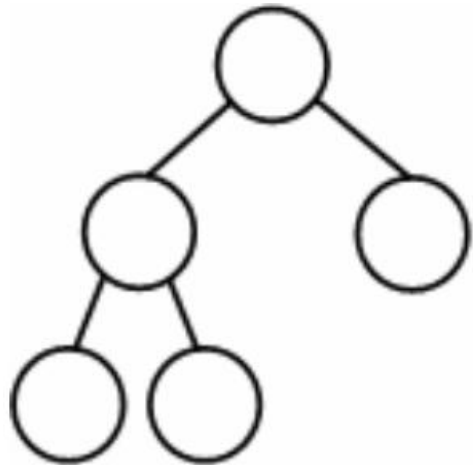
# Теоретичні відомості

Кожна *вершина* бінарного дерева є структурою, що складається з чотирьох видів полів.

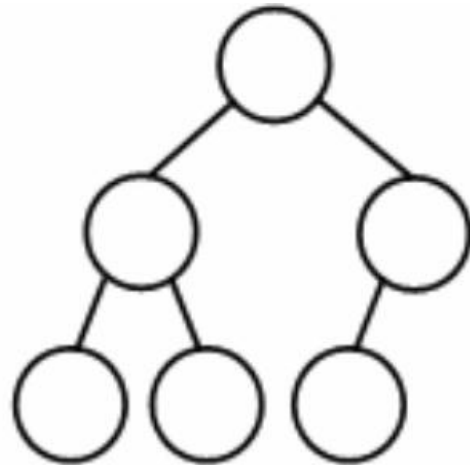
Вмістом цих полів будуть відповідно:

- інформаційне поле (ключ вершини);
- службове поле (їх може бути декілька або жодного);
- показчик на ліве піддерево;
- показчик на праве піддерево.

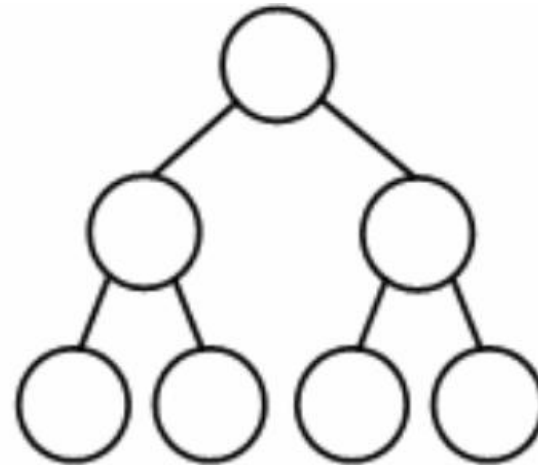
*Строге*



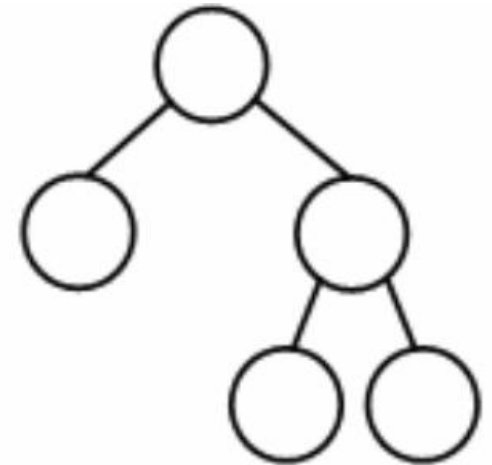
*Нестроге*



*Повне*



*Неповне*

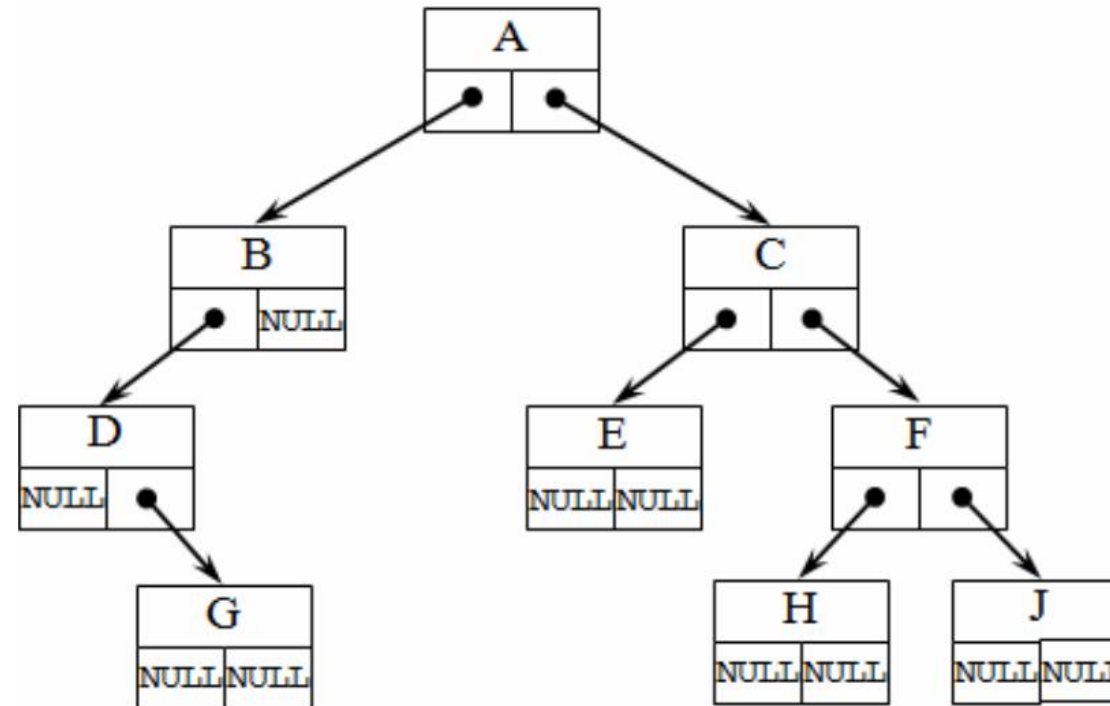


# Теоретичні відомості

Структура дерева відбивається у вхідному потоці даних таким чином: кожному порожньому зв'язку, який додається, відповідає умовний символ, наприклад, '\*' (зірочка).

При цьому спочатку описуються ліві нащадки, потім праві. Для структури бінарного дерева, показанного на рисунку, вхідний потік має вигляд:

ABD \* G \*\*\* CE \*\* FH \*\* J \*\*.



# Теоретичні відомості

---

Оголошення бінарного дерева виглядає таким чином:

```
public class ім'я типу
{
    Інформаційне поле;
    [Службове поле;]
    Адреса лівого піддерева;
    Адреса правого піддерева;
}
```

Наприклад:

```
public class Tree
{
    public int Data; // інформаційне поле
    public int count; // службове поле
    public Tree Left; // адреса лівого піддерева
    public Tree Right; // адреса правого піддерева
}
```



# Теоретичні відомості

---

Дамо де-які функції основних операцій при роботі з бінарним деревом.:

```
// Пошук елемента в бінарному дереві
public static Tree FindElement(int findData, Tree root)
{
    if (root == null || findData == root.Data)
        return root;
    if (root.Data > findData)
        return FindElement(findData, root.Left);
    else
        return FindElement(findData, root.Right);
}

// Мінімальний елемент дерева
public static Tree Minimum(Tree root)
{
    Tree l=root;
    while (l.Left != null)
    {
        l=l.Left;
    }
    return l;
}
```





# Дисципліна: “Програмування складних алгоритмів”

---

## Завдання до лабораторної роботи:

1. сформулювати математичну постановку задачі;
2. обрати метод розв’язання задачі, якщо це необхідно;
3. розробити графічну схему алгоритму;
4. записати розроблений алгоритм мовою програмування;
5. представити звіт до роботи.



# Завдання

| Варіант | Використовуваний спосіб обходу дерева для виконання завдання | Завдання   |
|---------|--|--|
| 1       | <i>послідовний</i>   | 1. Знайти суму всіх елементів дерева.<br>2. Вивести на екран всі листи дерева.   |
| 2       | <i>низхідний<br/>зворотній</i>                               | 1. Обчислити середнє арифметичне елементів дерева.<br>2. Вивести на екран ті листи дерева, які мають парні значення.     |
| 3       | <i>висхідний<br/>симетричний</i>                             | 1. Знайти кількість парних елементів дерева.<br>2. Вивести на екран значення вузлів, у яких тільки один нащадок (лівий). |
| 4       | <i>послідовний</i>   | 1. Обчислити кількість елементів дерева, кратних 10.<br>2. Видалити ті листи в дереві, у яких немає братів.              |
| 5       | <i>низхідний</i>   | 1. Збільшити всі елементи дерева в два рази.<br>2. Визначити, чи є дерево майже повним.                                  |



# Завдання

| Варіант | Використовуваний спосіб обходу дерева для виконання завдання | Завдання   |
|---------|--|--|
| 6       | <i>висхідний</i>   | 1. Всі від'ємні елементи дерева замінити на нулі.<br>2. Вивести на екран ті вузли, сума значень синів яких парна.  |
| 7       | <i>последовний</i>   | 1. Реверсувати кожен елемент дерева, тобто якщо є елемент 345, то замінюємо його на 543.<br>2. Визначити в якому поддереві (лівому або правому) кількість парних елементів більша.<br>Вивести ці елементи на екран для кожного піддерева окремо. |
| 8       | <i>низхідний</i>   | 1. Визначити кількість елементів дерева, що закінчуються на цифру 9<br>2. Вивести перелік тих вузлів, у яких кількість парних елементів в правому і лівому поддереві рівні.  |
| 9       | <i>последовний</i>   | 1. Ті елементи дерева, які складаються з однакових цифр, замінити значенням 0.<br>2. Вивести на екран ті вузли, у яких різниця сум елементів в правому і лівому поддереві більша 20.<br>Вивести ці суми для кожного вузла                        |

# Завдання

| Варіант | Використовуваний спосіб обходу дерева для виконання завдання | Завдання   |
|---------|--|--|
| 10      | <i>низхідний</i>   | 1. Знайти суму лівих елементів дерева.<br>2. Вивести на екран всі листи дерева.  |
| 11      | <i>последовний</i>   | 1. Обчислити середнє арифметичне елементів дерева.<br>2. Вивести на екран ті листи дерева, які мають непарні значення.   |
| 12      | <i>низхідний</i>   | 1. Знайти кількість парних елементів дерева.<br>2. Вивести на екран значення вузлів, у яких тільки один нащадок (лівий). |
| 13      | <i>последовний</i>   | 1. Обчислити кількість елементів дерева, кратних 5.<br>2. Видалити ті листи в дереві, у яких немає братів.               |
| 14      | <i>висхідний</i>   | 1.Збільшити всі елементи дерева в три рази.<br>2. Визначити, чи є дерево майже повним.                                   |





# Завдання

| Варіант | Використовуваний спосіб обходу дерева для виконання завдання | Завдання   |
|---------|--|--|
| 15      | <i>послідовний</i>   | 1. Всі від'ємні елементи дерева замінити на нулі.<br>2. Вивести на екран ті вузли, сума значень синів яких парна.  |
| 16      | <i>низхідний</i>   | 1. Реверсувати кожен елемент дерева, тобто якщо є елемент 345, то замінюємо його на 543.<br>2. Визначити в якому поддереві (лівому або правому) кількість парних елементів більша.<br>Вивести ці елементи на екран для кожного піддерева окремо. |
| 17      | <i>висхідний</i>   | 1. Визначити кількість елементів дерева, що закінчуються на цифру 3.<br>2. Вивести перелік тих вузлів, у яких кількість парних елементів в правому і лівому поддереві рівні.   |
| 18      | <i>послідовний</i>   | 1. Обчислити кількість елементів дерева, кратних 5.<br>2. Додати ті листи в дереві, у яких немає братів.   |
| 19      | <i>висхідний</i>   | 1. Збільшити всі елементи дерева на чотири.<br>2. Вивести на екран всі листи дерева.   |

# Завдання

| Варіант | Використовуваний спосіб обходу дерева для виконання завдання | Завдання   |
|---------|--|--|
| 20      | <i>послідовний</i>   | 1. Всі від'ємні елементи дерева замінити на нулі.<br>2. Вивести на екран ті вузли, сума значень синів яких не парна.   |
| 21      | <i>низхідний</i>   | 1. Реверсувати кожен елемент дерева, тобто якщо є елемент 234, то замінюємо його на 432.<br>2. Визначити в якому поддереві (лівому або правому) кількість парних елементів більша.<br>Вивести ці елементи на екран для кожного піддерева окремо. |
| 22      | <i>низхідний</i>   | 1. Визначити кількість елементів дерева, що закінчуються на цифру 4.<br>2. Вивести перелік тих вузлів, у яких кількість парних елементів в правому і лівому поддереві рівні.   |
| 23      | <i>послідовний</i>   | 1. Ті елементи дерева, які складаються з однакових цифр, замінити значенням 5.<br>2. Вивести на екран ті вузли, у яких різниця сум елементів в правому і лівому поддереві більша 8.<br>Вивести ці суми для кожного вузла                         |

# Завдання

| Варіант | Використовуваний спосіб обходу дерева для виконання завдання | Завдання  |
|---------|--|---|
| 24      | <i>низхідний</i>   | 1. Обчислити середнє арифметичне елементів дерева.<br>2. Вивести на екран ті листи дерева, які мають непарні значення.  |
| 25      | <i>последовний</i>   | 1. Знайти суму правих елементів дерева.<br>2. Вивести на екран всі листи дерева.  |
| 26      | <i>низхідний</i>   | 1. Знайти кількість не парних елементів дерева.<br>2. Вивести на екран значення вузлів, у яких тільки один нащадок (правий).  |
| 27      | <i>последовний</i>   | 1. Обчислити кількість елементів дерева, кратних 3.<br>2. Додати ті листи в дереві, у яких немає братів.  |
| 28      | <i>висхідний</i>   | 1.Збільшити всі елементи дерева на три.<br>2. Визначити в якому поддереві (лівому або правому) кількість парних елементів більша.<br>Вивести ці елементи на екран для кожного піддерева окремо. |

# Завдання

| Варіант | Використовуваний спосіб обходу дерева для виконання завдання | Завдання   |
|---------|--|--|
| 29      | <i>висхідний</i>   | <ol style="list-style-type: none"><li>1. Обчислити середнє арифметичне правих елементів дерева.</li><li>2. Вивести на екран ті листи дерева, які мають парні значення.</li></ol> |
| 30      | <i>послідовний</i>   | <ol style="list-style-type: none"><li>1. Знайти суму лівих елементів дерева.</li><li>2. Вивести на екран парні елементи дерева.</li></ol>  |

