Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

**Лабораторна робота №2**

**Завдання №2.1 – Біноміальна піраміда**

**Тип даних – Т9 (Дійсні числа)**

Виконав:

студент групи ІПС-23

факультету комп’ютерних наук

та кібернетики

Кирпотенко Юрій Володимирович

Київ 2023

**Зміст**

|  |  |
| --- | --- |
| Теоретичні відомості | 3 |
| Алгоритм | 3 |
| Складність | 4 |
| Мова програмування | 4 |
| Модулі програми | 5 |
| Інтерфейс користувача | 5 |
| Приклади | 6-7 |
| Висновок | 8 |
| Література | 8 |
|  |  |

**Теоретичні відомості**

Біноміальна піраміда - це структура даних, що представляється у вигляді піраміди, де кожен рівень містить числа, що отримуються з комбінації попереднього рівня чисел.

Така структура даних добре використовується для представлення біноміальних коефіцієнтів, які виникають у формулах біноміального розкладу та теорії ймовірностей. Крім того, біноміальні піраміди використовуються в різних областях математики та програмування, наприклад, для побудови алгоритмів пошуку та сортування.

**Алгоритм**

Алгоритм створення біноміального дерева з порядком k виглядає наступним чином:

1. Створюємо новий вузол, який стане коренем дерева з порядком k.
2. Якщо k = 0, повертаємо новий вузол як дерево з одним вузлом.
3. Якщо k > 0, то:

a. Створюємо першого нащадка кореня, який буде деревом з порядком k-1.

b. Рекурсивно створюємо нащадків кореня: другий нащадок буде деревом з порядком k-2, третій нащадок - деревом з порядком k-3 і т.д. до k-го нащадка, який буде деревом з порядком 0.

1. Повертаємо новий вузол з коренем, який має кількість нащадків, що дорівнює порядку дерева.

Цей алгоритм можна рекурсивно застосувати для створення біноміальної піраміди з довільною кількістю дерев.

**Складність**

Пошук має часову складність O(log n), де n - кількість вузлів у дереві. Якщо вузол з заданим ключем знаходиться в кореневому дереві, то часова складність буде O(1).

Вставка має часову складність O(log n), де n - кількість вузлів у дереві. Це через те, що вставка полягає у злитті двох біноміальних дерев зі степенем i, де i - номер рівня дерева. Якщо новий вузол додається до кореневого дерева, то часова складність буде O(1).

Видалення має часову складність O(log n), де n - кількість вузлів у дереві. Це через те, що при видаленні вузла здійснюється зв'язування дітей вузла у нове біноміальне дерево, а також перестановка цього дерева з іншими деревами з тим же степенем. Якщо вузол для видалення знаходиться у кореневому дереві, то часова складність буде O(log n), оскільки виконується пошук вузла, а потім його видалення.

**Мова програмування**

С# 11.0

Cередовище - Visual Studio 2022

**Інтерфейс користувача**

Вхідні данні задаються у коді програми.

Результат виводиться у консоль.

**Модулі програми**

***Class Program – точка входу в програму***

\* Оскільки цей клас відповідає за запуск алгоритму, у ньому ми отримуємо вхідні дані, обраховуємо піраміду, після чого виводимо результат.

static void Main(string[] args) – точка входу в програму

***Class BinomialHeap – біномальне дерево (піраміда)***

\* Клас-керувальник пірамідою та діями над нею.

public void Insert(T value) – вставка елементу

public (bool, BinomialTreeNode<T>?) Search(T value)public T DeleteMin() – пошук

public void Remove(T value) - видалення

**Приклади**

**Приклад використання**

1. **Введіть початкові дані у коді програми.**
2. **Оберіть необхідні дії у коді програми.**
3. **Запустіть програму.**
4. **Отримайте результат – відповідні результати згідно заданих умов.**

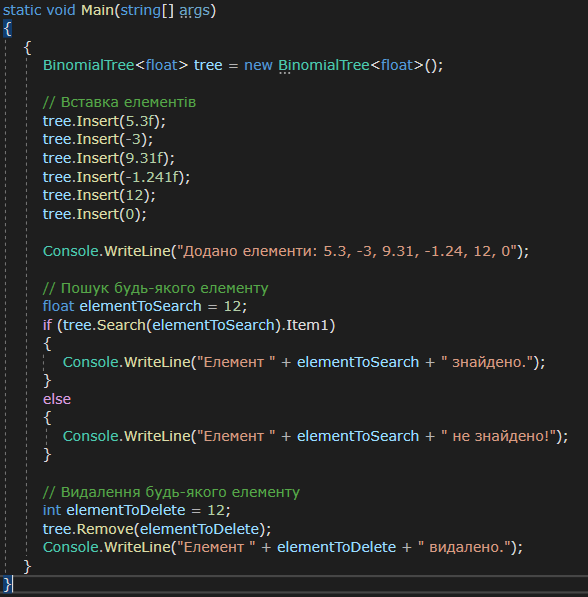
**Приклад з програми**

**Приклад 1**

**Створимо дерево та додамо дійсні числа**

**Знайдемо мінімальне та видалимо його**

**Знайдемо мінімальне ще раз**



**Отримаємо результат:**



**Висновок**

Програма може працювати з будь якою кілкістю строчок, це обмежено лиже можливостями приладу, на якому вона працює.

**Література**

<https://en.wikipedia.org/wiki/Pascal%27s_triangle>