Міністерство освіти і науки України
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА
Факультет прикладної математики та інформатики

Кафедра програмування

ЛАБОРАТОРНА РОБОТА № 9

Бітова Множина

з курсу “Алгоритми та структури даних”

Виконав:

Студент групи ПМІ-16

Бевз Маркіян Юрійович

Львів – 2024

Мета: Ознайомитись з бітовою множиною та розробити класу бітової множини, який дозволяє виконувати операції додавання, видалення, перевірки наявності та злиття елементів у бітовій множині. Здобуття навичок роботи з бітовими операціями та їх використання.

Принцип роботи бітової множини на основі бітового поля:

Додавання елементу (Add):

1. Визначається позиція біта, який представляє елемент у бітовому полі.
2. Біт на цій позиції встановлюється в 1, позначаючи присутність цього елемента в множині.

Видалення елементу (Remove):

1. Визначається позиція біта, який представляє елемент у бітовому полі.
2. Біт на цій позиції знімається, позначаючи відсутність цього елемента в множині.

Перевірка наявності елементу (Contains):

1. Визначається позиція біта, який представляє елемент у бітовому полі.
2. Перевіряється значення біта на цій позиції для визначення присутності або відсутності елемента в множині.

Злиття двох множин (Merge або Union):

1. Виконується побітова операція "або" над бітовими полями двох множин.
2. Отримане нове бітове поле представляє об'єднану множину, яка включає всі унікальні елементи обох множин.

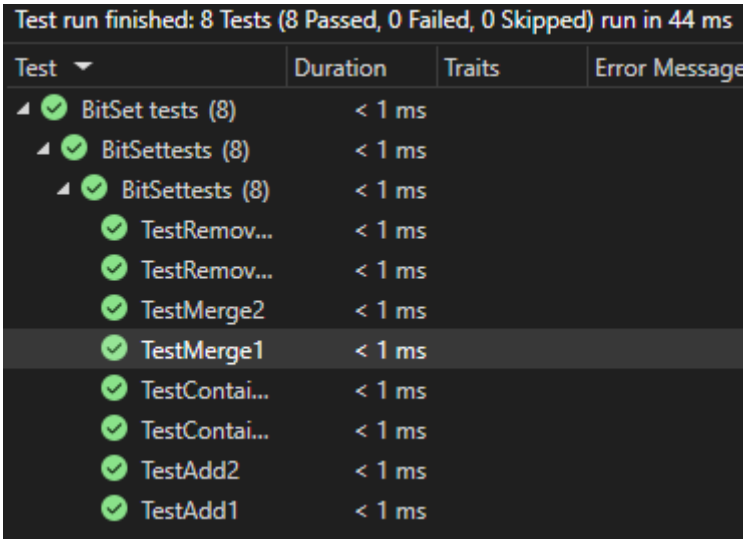
Очищення множини (Clear):

1. Всі біти в бітовому полі встановлюються в 0, щоб очистити множину. Такий підхід дозволяє ефективно використовувати бітові операції для реалізації операцій над множинами та забезпечує швидкий доступ до операцій в часі O(1).

Хід роботи:

Після того як я ретельно дослідив принцип роботи бітової множини на основі бітового поля реалізував клас BitSet для обробки цілих чисел. Написано методи додавання елемента в множину, вилучння, перевірки на наявність та злиття двох множин. Також створено 8 тестів, що перевіряють коректність виконання методів, нижче буде подано зображення виконання програми та зображення й код тестів.

```
Basic set:
00000000000000000000000000000000
Added two elements:
00000000000000000000000010100000
Contains 5: true
Contains 7: true
00000000000000000000000010000000
Contains 5: false
Contains 7: true
Set 1: 00000000000000000000000000101010
Set 2: 00000000000000000000000001011000
Merged Set: 00000000000000000000000001111010
```




```
TEST_CLASS(BitSettests)
{
public:
    TEST_METHOD(TestAdd1)
    {
        BitSet mySet;
        mySet.add(2);
        Assert::IsTrue(mySet.contains(2), L"Failed to add element 2");
    }
    TEST_METHOD(TestAdd2)
    {
        // Тест 2: Додаємо елемент 5 (позначаємо 6-й біт)
        BitSet mySet;
        mySet.add(5);
        Assert::IsTrue(mySet.contains(5), L"Failed to add element 5");
    }
    TEST_METHOD(TestRemove1)
    {
        BitSet mySet;
        //даємо елемент 3, потім видаляємо його
        mySet.add(3);
        mySet.remove(3);
        Assert::IsFalse(mySet.contains(3), L"Failed to remove element 3");
    }
```

```
TEST_METHOD(TestRemove2)
{
    BitSet mySet;
    // Тест 2: Додаємо елемент 0, потім видаляємо його
    mySet.add(0);
    mySet.remove(0);
    Assert::IsFalse(mySet.contains(0), L"Failed to remove element 0");
}

TEST_METHOD(TestContains1)
{
    BitSet mySet;
    // Тест 1: Перевіряємо, чи множина містить доданий елемент
    mySet.add(4);
    Assert::IsTrue(mySet.contains(4), L"Contains method failed for element 4");
}

TEST_METHOD(TestContains2)
{
    BitSet mySet;

    // Тест 2: Перевіряємо, чи множина не містить елемент, який не був доданий
    Assert::IsFalse(mySet.contains(7), L"Contains method failed for non-existing element 7");
}
TEST_METHOD(TestMerge1)
{
    BitSet set1, set2;
    // Тест 1: Злиття порожніх множин
    BitSet mergedSet1 = set1.merge(set2);
    Assert::IsTrue(mergedSet1.contains(0) == false, L"Merged set is not empty");

}

TEST_METHOD(TestMerge2)
{
    BitSet set1, set2;
    // Тест 2: Злиття множин з однаковими елементами
    set1.add(2);
    set1.add(4);
    set2.add(4);
    set2.add(6);
    BitSet mergedSet2 = set1.merge(set2);
    Assert::IsTrue(mergedSet2.contains(2), L"Merged set doesn't contain element 2");
    Assert::IsTrue(mergedSet2.contains(4), L"Merged set doesn't contain element 4");
    Assert::IsTrue(mergedSet2.contains(6), L"Merged set doesn't contain element 6");
}
```

Висновок: Я ознайомився з бітовою множиною та розробив клас бітової множини, який дозволяє виконувати операції додавання, видалення, перевірки н аявності та злиття елементів у бітовій множині. Здобув навички роботи з бітови ми операціями та навчився їх використовувати.