**БГУИР**

**Кафедра ИИТ**

**Отчет по лабораторной работе №1**

**Тема: «Алгоритмы операций над множествами и их реализация»**

Выполнил:

Студент группы 421702 Мяло В. В.

Проверил:

Рудьман И. О.

Минск

2024

**Цель работы:**

Научиться составлять алгоритм выполнения операций над множествами.

**Задача:**

Реализовать алгоритм на языке программирования C++, выполняющий следующие операции над множествами:

1. Объединение

2. Пересечение

3. Разность

4. Симметрическая разность

5. Дополнение до универсума.

**Уточнение постановки задачи:**

Множества в реализации данного алгоритма будут задаваться при помощи динамического массива (для этих целей будет использоваться контейнер vector). Мощности множеств будут задаваться пользователем. Универсальное множество (универсум) будет содержать целые числа диапазона [1; N]. Сами множества, вводимые пользователями также будут содержать целые числа из диапазона [1; N].

**Определения и понятия, использовавшиеся в ходе выполнения работы:**

1. Объединением, двух множеств X и Y называется множество, обозначаемое XUY и состоящее из элементов, принадлежащих хотя бы одному из множеств X или Y.

2. Пересечением, множеств X и Y называется множество, обозначаемое X∩Y состоящее из элементов, принадлежащих каждому из множеств X и Y.

3. Разностью множеств X и Y называется множество, обозначаемое X\Y и

состоящее из всех элементов X, не принадлежащих Y.

4. Симметричной разностью множеств X и Y называется множество:

X∆Y = (X\Y)U(Y\X).

5. Дополнением к множеству X относительно универсального множества U называется множество X’ = U\X

**Функции для упрощения реализации программы:**

Для более простой реализации алгоритмов я реализовал и использовал следующие функции:

1. void vector\_vivod(vector<double> a) - функция, выводящая множество. Выводит на экран множество.

2. vector<double> raznost(vector<double> a, vector<double> b) - функция, находящая разность двух множеств. Возвращает разность двух множеств в виде массива.

3. vector<double> obyedineniye(vector<double> a, vector<double> b) - функция, находящая объединение двух множеств. Возвращает объединение двух множеств в виде массива.

4. bool vector\_is\_uniq(vector<double> a) - функция, проверяющая множества на наличие повторяющихся элементов. Возвращает “True”, если в множестве отсутствуют повторяющиеся элементы, и возвращает “False”, если в множестве есть повторяющиеся элементы.

**Алгоритм реализации операций:**

1. Операция пересечения множеств X и Y. Алгоритм следующий:

1.1. Создаем пустое множество V, в которое будет помещен результат выполнения операции пересечения.

1.2. Берем первый элемент множества X, переходим к пункту 1.4.

1.3. Берем следующий элемент множества X.

1.3.1. Если достигли конца множества, конец операции.

1.4. Берем первый элемент множества Y, переходим к пункту 1.6.

1.5. Берем следующий элемент множества Y.

1.5.1. Если достигли конца множества, переходим к пункту 1.3.

1.6. Сравниваем взятые элементы.

1.6.1. Если они равны, то добавляем этот элемент в множество V, переходим к пункту 1.3.

1.6.2. Если они не равны, то переходим к пункту 1.5.

2. Операция объединения множеств X и Y. Алгоритм следующий:

2.1. Создаем пустое множество V, в которое будем помещен результат выполнения операции объединения.

2.2. Заполняем множество V всеми элементами множества X.

2.3. Берем первый элемент множества Y, переходим к пункту 2.5.

2.4. Берем следующий элемент множества Y, переходим к пункту 2.5.

2.4.1. Если достигли конца множества, конец операции.

2.5. Берем первый элемент множества X, переходим к пункту 2.7.

2.6. Берем следующий элемент множества X.

2.6.1. Если достигли конца множества:

2.6.1.1. Добавляем взятый из множества Y элемент в результирующее множество.

2.6.1.2. Переходим к пункту 2.4.

2.7. Сравниваем взятые элементы.

2.7.1. Если элементы разные, переходим к пункту 2.6.

2.7.2. Если элементы одинаковые, переходим к пункту 2.4.

3. Разность множеств X и Y (X/Y).

3.1. Создаем пустое множество V, в которое будет помещен результат выполнения операции.

3.2. Берем первый элемент множества X, переходим к пункту 3.4.

3.3. Берем следующий элемент множества X.

3.3.1. Если достигли конца множества, конец операции.

3.4. Берем первый элемент множества Y, переходим к пункту 3.6.

3.5. Берем следующий элемент множества Y.

3.5.1. Если достигли конца множества:

3.5.1.1. Добавляем взятый элемент X в множество V.

3.5.1.2. Переходим к пункту 3.3.

3.6. Сравниваем взятые элементы.

3.6.1. Если они равны, то переходим к пункту 3.3.

3.6.2. Если они не равны, то переходим к пункту 3.5.

4. Симметрическая разность множеств X и Y.

4.1. Находим разность множеств X и Y (X/Y). (см. пункт 3)

4.2. Находим разность множеств Y и X (Y/X) . (см. пункт 3, меняем X и Y местами)

4.3. Объединяем два полученных множества. (см. пункт 2)

5. Дополнение множества X или Y по множеству U.

5.1. Находим разность множеств U и X (U/X) или U и Y (U/Y) соответственно. (см. пункт 3)

**Алгоритм выполнения программы:**

1. Сначала пользователь вводит больший член множества U (по совместительству мощность U).

2. Пользователь вводит мощность множества X.

2.1. Если введенная мощность больше N, переходим к пункту 2.

3. Пользователь вводит элементы множества X.

3.1. Если введенный элемент не входит в множество U, пользователь получает соответствующее сообщение и получает возможность ввести элемент заново.

3.2. Если введенное множество имеет повторяющиеся элементы, переходим к пункту 3.

4. Пользователь вводит мощность множества Y.

4.1. Если введенная мощность больше N, переходим к пункту 4.

5. Пользователь вводит элементы множества Y.

5.1. Если введенный элемент не входит в множество U, пользователь получает соответствующее сообщение и получает возможность ввести элемент заново.

5.2. Если введенное множество имеет повторяющиеся элементы, переходим к пункту 5.

6. Далее проводятся операции над множествами, в консоль последовательно выводятся ответы. (см. пункт “Алгоритмы реализации операций”)

**Вывод:**

В ходе выполнения данной лабораторной работы я научился составлять алгоритм выполнения операции над множествами и сделал это, используя язык программирования C++.