Министерство образования Республики Беларусь

Учреждение образования

<<Белорусский государственный университет информатики и радиоэлектроники>>

Кафедра интеллектуальных информационных технологий

Лабараторная работа №1

Алгоритмы операций над множествами и их реализация

Выполнил:студент гр.421702

Сушко В.Р.

Проверил:Рудьман И.О.

Минск 2024

Цель: научиться составлять алгоритм выполнения операций над множествами.

Постановка задачи

Задача: Выполнить базовые операции над множествами

Давайте разберем алгоритм операции объединения двух массивов на C++ по шагам. Суть операции — это объединить два массива, добавив элементы второго массива в первый только если они отсутствуют в первом. Это делается для того, чтобы избежать дублирования элементов в результате.

### Подробное объяснение алгоритма

1. \*\*Инициализация\*\*: Вы имеете два множества: А и `Б, длины которых определяются переменными `а` и `б`.

2. \*\*Вывод элементов первого массива\*\*:

- Мы берем элемент множества А.

-Выводим его.

-Переходим к следующему

-Повторяем действия пока мы не дойдем до конца множества

3. \*\*Проверка и вывод уникальных элементов второго массива\*\*:

- Мы берем первый элемент множества Б .

-Мы берем первый элемент во множестве А

-Сравниваем элемент Б и элемент А

-Если они различны,то повторяем эти действия со следующим элементом

-Если они одинаковые,то завершаем сравнение и выводим,что условие неверно

-Если условие верно переходим к следующему пункту,если нет,то возвращаемся к первому пункту и переходим к следующему элементу множества А

-Выводим его.

-Переходим к следующему элементу множества Б

-Повторяем действия пока мы не дойдем до конца множества

### Описание алгоритма пересечения:

1. \*\*Инициализация\*\*:

- У вас есть два множества, которые мы будем называть А и Б.

- Их размеры обозначаются переменными а для первого массива и `б` для второго.

2. \*\*Перебор элементов первого массива\*\*:

- Мы берем первый элемент множества А .

-Мы берем первый элемент во множестве Б

-Сравниваем элемент Б и элемент А

-Если они одинаковые,то выводим,что условие верноя

-Если они различные,то переходим к следующему элементу множества Б

-Если условие верно переходим к следующему пункту,если нет,то возвращаемся к первому пункту и переходим к следующему элементу

-Выводим его.

-Переходим к следующему.

### Итог:

Алгоритм позволяет найти и вывести все элементы, которые присутствуют в обоих массивах. Это полезно, когда вам нужно определить общие элементы между двумя наборами данных.

#Разность относительно первого множества (второго аналогично)

1. \*\*Инициализация цикла\*\*:

- Мы берем элемент множества А.

-Мы берем первый элемент во множестве Б

-Сравниваем элемент Б и элемент А

-Если они одинаковые,то выводим,что условие неверно

-Если они различные,то переходим к следующему элементу множества Б

-Если мы дошли до конца множества и не вывели,что условие неверно,выводим условие верно

-Если условие верно переходим к следующему пункту,если нет,то переходим к следующему элементу

-Выводим его.

-Переходим к следующему.

-Повторяем действия пока мы не дойдем до конца множества

### Итог

Таким образом, данный фрагмент кода проходит по всем элементам первого массива А, проверяет, есть ли каждый элемент во втором массиве Б и выводит только те элементы из массива А которые отсутствуют в массиве Б. Это полезно для выявления уникальных элементов в первом массиве.

###Симметричная разность

1. \*\*Первая часть: проверка первого массива (А)\*\*:

-Мы берем первый элемент множества А.

-Мы берем первый элемент во множестве Б

-Сравниваем элемент Б и элемент А

-Если они одинаковые,то выводим,что условие неверно

-Если они различные,то переходим к следующему элементу множества Б

-Если мы дошли до конца множества и не вывели,что условие неверно,выводим условие верно

-Если у ловие верно переходим к следующему пункту,если нет,то переходим к следующему элементу

-Выводим его.

-Переходим к следующему.

-Повторяем действия пока мы не дойдем до конца множества

2. \*\*Вторая часть: проверка второго массива (`Б`)\*\*:

- Мы берем первый элемент множества Б.

-Мы берем первый элемент во множестве А

-Сравниваем элемент Б и элемент А

-Если они одинаковые,то выводим,что условие неверно

-Если они различные,то переходим к следующему элементу множества А

-Если мы дошли до конца множества и не вывели,что условие неверно,выводим условие верно

-Если условие верно переходим к следующему пункту,если нет,то переходим к следующему элементу

-Выводим его.

-Переходим к следующему.

-Повторяем действия пока мы не дойдем до конца множества

### Итог

Таким образом, данный фрагмент кода выполняет поиск уникальных элементов в обоих массивах:

- Он сначала находит и выводит все уникальные элементы из А которые не присутствуют в Б.

- Затем он находит и выводит все уникальные элементы из Б которые не присутствуют в А.

Этот подход позволяет собрать все элементы, которые присутствуют только в одном из двух множеств, и выдать итоговый список уникальных значений.

###Универсум

- Мы берем первый элемент множества В.

-Мы берем первый элемент во множестве Б(А)

-Сравниваем элемент В и элемент Б(А)

-Если они одинаковые,то выводим,что условие неверно

-Если они различные,то переходим к следующему элементу множества Б

-Если мы дошли до конца множества и не вывели,что условие неверно,выводим условие верно

-Если условие верно переходим к следующему пункту,если нет,то переходим к следующему элементу

-Выводим его.

-Переходим к следующему.

-Повторяем действия пока мы не дойдем до конца множества

### Итог

Таким образом, этот код выполняет следующую задачу:

- Он проходит по массиву В и находит элементы, которые отсутствуют в массиве А(Б)

- Уникальные элементы из В, не присутствующие в А(Б)выводятся на экран.

Этот подход позволяет быстро идентифицировать элементы массива В которые не включены в А.

###Вывод программы

Vvedite dlinu 1 mnogestva 3

Vvedite dlinu 2 mnogestva 3

Zapolnite 1 mnogestvo

Universom ot 1 do 10

1

2

3

Zapolnite 2 mnogestvo

3

4

5

Obedenenie: 1 2 3 4 5

Peresechenie: 3

Raznost1: 1 2

Raznost2: 4 5

Simmetria: 1 2 4 5

Universe1: 4 5 6 7 8 9 10

Universe2: 1 2 6 7 8 9 10

###Вывод лабараторной работы №1

Выполняя эту работу я научился составлять алгоритмы операций над множествами:объединение,пересечение,разность,универсум.Все алгоритмы операций были выполнены на одном из предложенных языков – C++.