Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

за 1 семестр

По дисциплине: «Дискретная математика»

Тема: «Множества»

Выполнил:

Студент 2 курса

Группы ПО-4(2)

Кречко К. А.

Проверила:

Глущенко Т.А.

2020

ЛАБОРАТОРНАЯ РАБОТА №1. Множества

**Вариант 14**

**Задание 1.**

На универсуме U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11} заданы множества A, B, C



1. Для заданного множества A построить булеан.

**Код**:

#include<iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Множество А={ ";

int A[] = { 1,3,5,6,7,8 };

int size = sizeof(A) / sizeof(int);

for (int i = 0; i < size; i++)

{

cout << A[i] << " ";

}

cout << "}" << endl << "Булеан: " << endl;

cout << " {";

for (int i = 0; i < pow(2, size); i++) {

cout << " {";

for (int j = 0, num = i; num != 0; j++) {

if (num % 2 == 1) {

cout << A[j] << ", ";

}

num /= 2;

}

cout << '\b' << '\b' << "},";

if (i != pow(2, size) - 1) {

cout << endl;

}

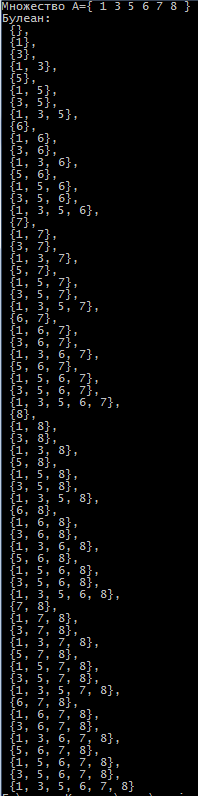
}

cout << '\b';

return 0;

}

**Результат:**



2. На выбор реализовать сортировку слиянием либо алгоритмом «слияние» построить объединение множеств A и B.

**Код:**

#include<iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int A[] = { 1,3,5,6,7,8 };

int B[] = { 2,3,4,5,8 };

int sizeA = sizeof(A) / sizeof(int);

int sizeB = sizeof(B) / sizeof(int);

int aindex = 0;

int bindex = 0;

int cindex = 0;

// создание массива для отсортированной последовательности

int\* c = new int[sizeA + sizeB];

// сортировка

while ((aindex < sizeA) && (bindex < sizeB))

{

if (A[aindex] < B[bindex])

{

c[cindex] = A[aindex];

aindex++;

}

else

{

c[cindex] = B[bindex];

bindex++;

}

cindex++;

}

// загружаем остатки строк (работать будет всегда только один из этих циклов)

while (aindex < sizeA)

c[cindex++] = A[aindex++];

while (bindex < sizeB) // загружаем остатки строк

c[cindex++] = B[bindex++];

// вывод получившейся последовательности

for (int i = 0; i < cindex; i++) {

if (c[i] == c[i - 1]) //если элемент повторяется - пропускаем его

continue;

cout << c[i] << " ";

}

cin.get();

return 0;

}

**Результат: **

3. Реализовать операцию пересечения множеств, используя маску.

**Код:**

#include<iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int U[] = { 1,2,3,4,5,6,7,8,9,10,11 };

int size\_U = sizeof(U) / sizeof(int);

int A[] = { 1,3,5,6,7,8 };

int B[] = { 2,3,4,5,8 };

int sizeA = sizeof(A) / sizeof(int);

int sizeB = sizeof(B) / sizeof(int);

int\*\* mask = new int\* [2];

for (int i = 0; i<2; i++) {

mask[i] = new int[size\_U];

}

for (int i = 0, j = 0, k = 0; i < size\_U; i++) {

if (U[i] == A[i - j]) {

mask[0][i] = 1;

}

else {

mask[0][i] = 0;

j++;

}

if (U[i] == B[i - k]) {

mask[1][i] = 1;

}

else {

mask[1][i] = 0;

k++;

}

}

cout<<"{";

for (int i = 0; i < size\_U; i++) {

if (mask[0][i] & mask[1][i]) {

cout<<U[i]<<", ";

}

}

cout << '\b' << '\b' << "}";

for (int i = 0; i <2; i++) {

delete[] mask[i];

}

delete[] mask;

return 0;

}

**Результат:**

4. Найти прямое произведение A×B и его мощность.

**Код:**

#include<iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int A[] = { 1,3,5,6,7,8 };

int B[] = { 2,3,4,5,8 };

int sizeA = sizeof(A) / sizeof(int);

int sizeB = sizeof(B) / sizeof(int);

for (int i = 0; i < sizeA; i++) {

for (int j = 0; j < sizeB; j++) {

cout << "(" << A[i] << "," << B[j] << "), ";

}

cout << endl;

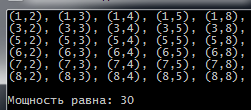
}

cout <<"\nМощность равна: " << sizeA \* sizeB << endl;

return 0;

}

**Результат:**

****

5. Вычислить программно выражение согласно варианту, проиллюстрировать результат диаграммой Эйлера-Венна.

**Код:**

#include<iostream>

using namespace std;

int Foo(int B[], int C[], int &k, int sizeB, int sizeC) { ... }

{

for (int i = 0; i < sizeB; i++) {

for (int j = 0; j < sizeC; j++) {

if (B[i] == C[j]) {

break;

}

if ((j + 1) == sizeC) {

k++;

}

}

}

return k;

}

int main()

{

setlocale(LC\_ALL, "ru");

int A[] = { 1,3,5,6,7,8 };

int B[] = { 2,3,4,5,8 };

int C[] = { 1,2,3,5,6 };

int sizeA = sizeof(A) / sizeof(int);

int sizeB = sizeof(B) / sizeof(int);

int sizeC = sizeof(C) / sizeof(int);

int kol = 0;

Foo(B, C, kol, sizeB, sizeC); // функция, которая считает кол-во элементов (B/C)

int\* arr1 = new int[kol];

int a = 0;

for (int i = 0; i < sizeB; i++) {

for (int j = 0; j < sizeC; j++) {

if (B[i] == C[j]) {

break;

}

if (j + 1 == sizeC) {

arr1[a] = B[i];

a++;

}

}

}

cout << "B-C= ";

for (int i = 0; i < kol; i++)

{

cout << arr1[i] << ",";

}

cout << endl;

int kol1 = 0;

Foo(A, arr1, kol1, sizeA, kol); // функция, которая считает кол-во элементов A/(B/C)

int\* arr2 = new int[kol1];

a = 0;

for (int i = 0; i < sizeA; i++) {

for (int j = 0; j < kol; j++) {

if (A[i] == arr1[j]) {

break;

}

if (j + 1 == kol) {

arr2[a] = A[i];

a++;

}

}

}

cout << "A-(B-C)= ";

for (int i = 0; i < kol1; i++)

{

cout << arr2[i] << ",";

}

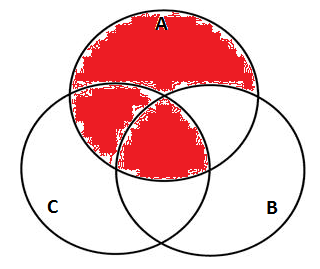
delete[] arr1;

delete[] arr2;

return 0;}

**Результат: **

**Диаграмма Эйлера-Венна:**

****

6. Указать примеры покрытий и разбиения множества A:

А={1,3,5,6,7,8}

Примеры покрытий:

E1= {{1,3},{3,5,6,7,8}};

E2= {{1,3,5},{3,5,6,7,8}};

Е3= {{1,8},{1,7},{1,6},{1,3,5}};

Е4= {{5,6},{3,5,6,7},{1,3,5,6,7,8}};

Е5= {{1},{1,3},{1,3,5},{1,3,5,6,7,8}};

Примеры разбиения:

E1= {{1,3,5,6,7,8}};

E2= {{1},{3,5,6,7,8}};

Е3= {{5},{1,3,6,7,8}};

Е4= {{3,6},{1,5,7,8}};

Е5= {{7,8},{1,3,5,6}};

Е6= {{1,3},{5,6},{7,8}};

Е7={{1,3,5},{6,7,8}};

Е8= {{1,6,8},{3,5,7}};

Е9= {{1},{3},{5},{6},{7},{8}};

8. Построить для вашего множества (или множества меньшей мощности) бинарный код Грея.

**Код:**

#include<iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int A[] = { 1,3,5,6,7,8 };

int sizeA = sizeof(A) / sizeof(int);

int\* gray\_code = new int[sizeA];

for (int i = 0; i < sizeA; i++) {

cout << A[i] <<" ";

A[i] = A[i] ^ (A[i] >> 1);

for (int j =3; j >= 0; j--) {

gray\_code[j] = A[i] % 2;

A[i] /= 2;

}

for (int j = 0; j < 4; j++) {

cout << gray\_code[j];

}

cout << endl;

}

delete[] gray\_code;

return 0;

}

**Результат:**

****

9. Составить таблицу для различных сортировок (быстрой, слиянием, выбором, вставки, пятая на выбор) с указанием «О большого», достоинств и недостатков данного алгоритма сортировки.

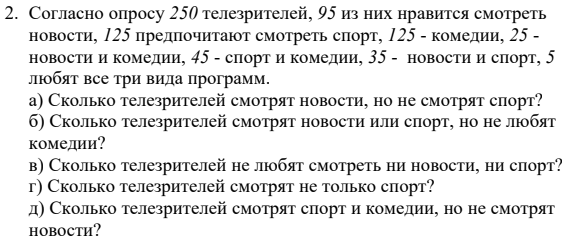


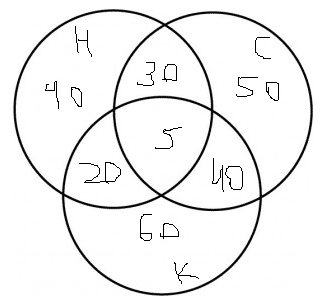
**Задание 2.**

Решить задачу на применение формулы включений и исключений согласно варианту. Вариант задания равен остатку от деления вашего порядкого номера на 3.

14/3=4 (ост. 2)

**Вариант 2**

****

****

а). 95-30-5=60;

б). 40+50=90;

в). 60;

г). 75;

д). 40;

**Вывод:** Изучил множества и действия с ними. Узнал что такое булеан, маска и сортировка слиянием. Изучил бинарный код Грея.