Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій



Звіт

про виконання лабораторної роботи №2

Множини

з дисципліни «Дискретна математика»

студента IT-11 Лісничука Арсена

2021

Варіант 13

Мета:

Вивчення множин, операцій над множинами, комп’ютерного подання множин.

Завдання 1:

Умова:

U {1, 2, …, 24,25}

A {1,2,3,4,13,14,16,17,21}

B {1,3,4,5,6,13,14,15,20,21}

C {1,9,10,11,12,13,18,19,24}

(А∪С) \ С ∪ В) \

Текст програми:

#include <iostream>

#include <algorithm>

#include <cmath>

using namespace std;

#define GET\_ARRAY\_LENGHT \_countof

int UniversalPluralSize;

int\* UniversalPlural;

// input

int A[] = { 1, 2, 3, 4, 13, 14, 16, 17, 21 };

const int SizeA = GET\_ARRAY\_LENGHT(A);

int B[] = { 1, 3, 4, 5, 6, 13, 14, 15, 20, 21 };

const int SizeB = GET\_ARRAY\_LENGHT(B);

int C[] = { 1, 9, 10, 11, 12, 13, 18, 19, 24 };

const int SizeC = GET\_ARRAY\_LENGHT(C);

// from = -100, to = 200 = {-100, -99, -98 ... 198, 199, 200 }

void SetUniversalPlural(const int from, const int to)

{

UniversalPluralSize = to - from + 1;

UniversalPlural = new int[UniversalPluralSize];

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

{

UniversalPlural[i] = from + i;

}

}

// Print array

void PrintPluar(const int\* p, const int size, const char\* addString = nullptr)

{

if (addString != nullptr)

{

cout << "[" << addString << "] ";

}

cout << "{";

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

{

cout << p[i];

if (i != size - 1)

{

cout << ", ";

}

}

cout << "}" << endl;

}

// A \ B

// {0, 4, -1, 5} \ {2, 3, 5, -2, 0} = {4, -1}

int\* Divide(const int\* a, const int size\_a, const int\* b, const int size\_b, int& size, bool isSort = true)

{

int index = 0;

int\* tmp = new int[size\_a];

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

{

for (int j = 0; j < size\_b; j++)

{

if (a[i] == b[j])

{

tmp[index] = i;

index++;

}

}

}

size = size\_a - index;

int\* res = new int[size];

for (int i = 0, i1 = 0; i < size\_a; i++)

{

if (i == tmp[i1])

{

i1++;

}

else

{

res[i - i1] = a[i];

}

}

delete[] tmp;

tmp = nullptr;

if (isSort)

sort(res, res + size);

return res;

}

// A + B

// {0, 1, 2, 3} + {5, 8, 0, 1, 2} = {0, 1, 2, 3, 5, 8, 0, 1, 2}

int\* Add(const int\* a, const int size\_a, const int\* b, const int size\_b, int& size, bool isSort = true)

{

size = size\_a + size\_b;

int\* res = new int[size];

int total = 0;

for (int i = 0; i < size\_a; i++, total++)

{

res[i] = a[i];

}

for (int i = 0; i < size\_b && total < size; i++, total++)

{

res[total] = b[i];

}

if (isSort)

sort(res, res + size);

return res;

}

// A u B

// {0, -1, -3, 5, 4, 9} u {3, 2, 5, 6, 7} = {-3, -1, 0, 2, 3, 4, 5, 6, 7, 9 }

int\* Unification(const int\* a, const int size\_a, const int\* b, const int size\_b, int& size, bool isSort = true)

{

int\* tmp = new int[size\_b];

int indexTmpArray = 0;

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

{

for (int j = 0; j < size\_b; j++)

{

if (a[i] == b[j])

{

tmp[indexTmpArray] = a[i];

indexTmpArray++;

}

}

}

int\* tmp1 = new int[indexTmpArray];

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

{

tmp1[i] = tmp[i];

//cout << tmp[i] << " ";

}

delete[] tmp;

tmp = nullptr;

int s;

int\* tmp2 = Divide(b, size\_b, tmp1, indexTmpArray, s);

delete[] tmp1;

int\* res = Add(tmp2, s, a, size\_a, size);

if (isSort)

sort(res, res + size);

return res;

}

// !A

// {1, 3, 5, 7, 14} = { 2, 4, 6, 8, 9, 10, 11, 12, 13, 15, ... , 25 }

int\* Not(const int\* a, const int size\_a, int& size, bool isSort = true)

{

return Divide(UniversalPlural, UniversalPluralSize, a, size\_a, size);

}

// ((A u C)\C u B)\(!C)

int main()

{

// U[25]

SetUniversalPlural(1, 25);

PrintPluar(UniversalPlural, UniversalPluralSize, "U");

PrintPluar(A, SizeA, "A");

PrintPluar(B, SizeB, "B");

PrintPluar(C, SizeC, "C");

cout << endl;

// A u C

int size\_p1;

int\* p1 = Unification(A, SizeA, C, SizeC, size\_p1);

PrintPluar(p1, size\_p1, "A u C");

// !C

int size\_p2;

int\* p2 = Not(C, SizeC, size\_p2);

PrintPluar(p2, size\_p2, "!C");

// (A u C) \ C

int size\_p3;

int\* p3 = Divide(p1, size\_p1, C, SizeC, size\_p3);

PrintPluar(p3, size\_p3, "(A u C) \\ C");

// ((A u C) \ C) u B

int size\_p4;

int\* p4 = Unification(p3, size\_p3, B, SizeB, size\_p4);

PrintPluar(p4, size\_p4, "((A u C) \\ C) u B");

// (((A u C) \ C) u B) \ (!C)

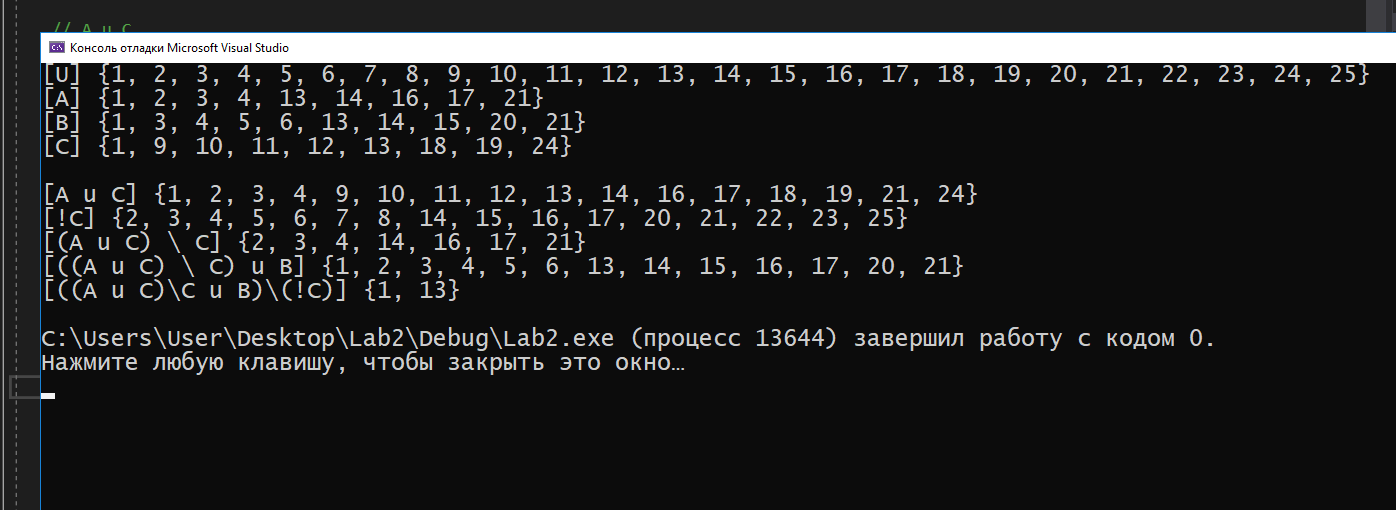
int size\_p5;

int \*p5 = Divide(p4, size\_p4, p2, size\_p2, size\_p5);

PrintPluar(p5, size\_p5, "((A u C)\\C u B)\\(!C)");

return 0;

}

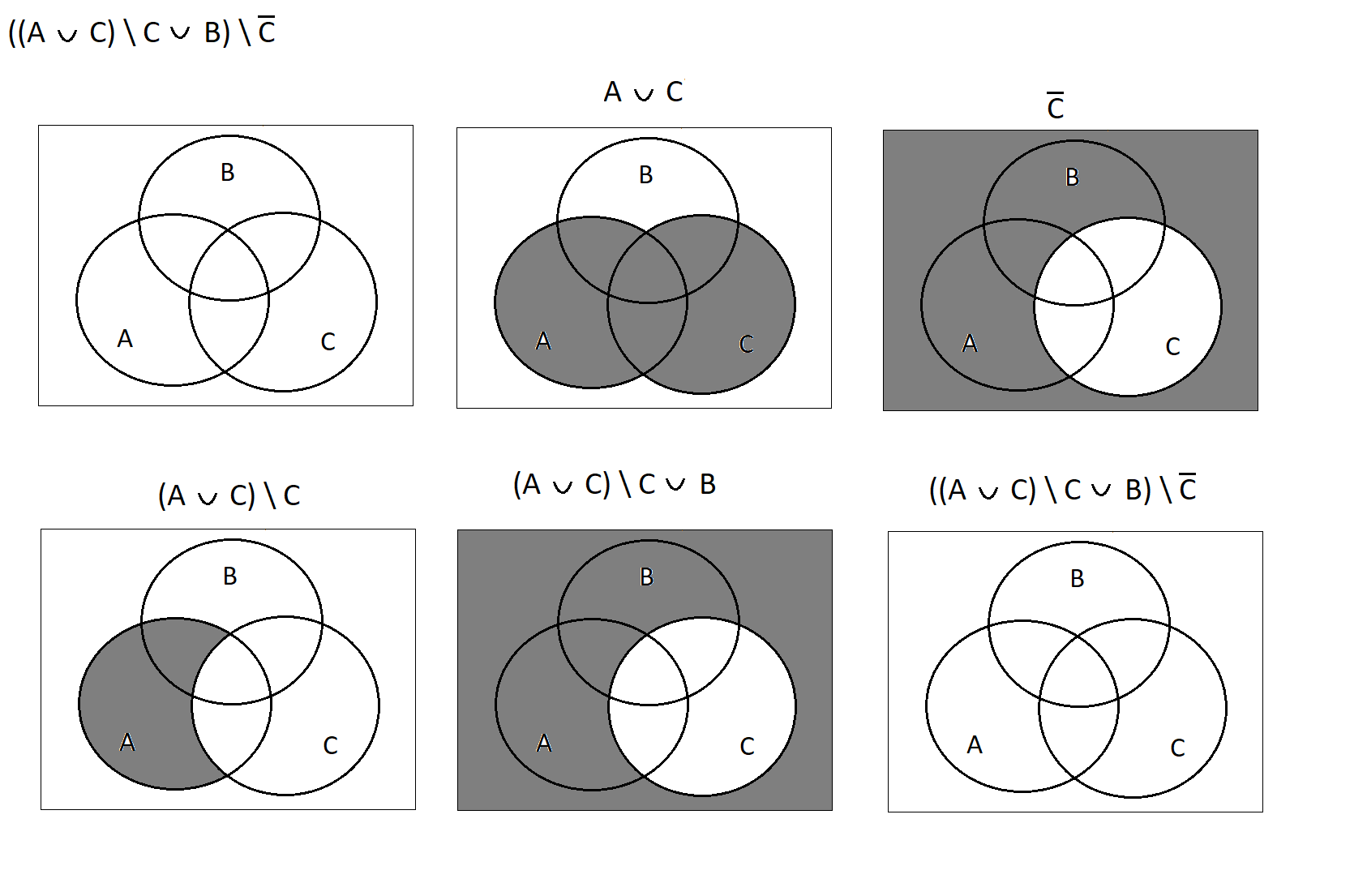


Завдання 2:

Умова:

Для формул, наведених у табл. 3, побудувати діаграми Венна за допомогою будь-якого графічного редактора.

Розвязок:



Висновок: В цій лабораторній роботі я навчився працювати із множинами.