

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
Санкт-Петербургский государственный  
электротехнический университет «ЛЭТИ» имени В.И. Ульянова (Ленина)

Факультет компьютерных технологий и информатики  
Кафедра вычислительной техники

Зачётная работа №1  
по дисциплине «Алгоритмы и структуры данных»  
«Множества»

Вариант №21

Выполнил: студент группы 5005  
Зинатуллин Азат Салаватович

Проверил: старший преподаватель  
Колинько Павел Георгиевич

## 1. Цель работы

Научиться представлять множества разными способами и оценить временную сложность алгоритмов для каждого способа, а так же сравнить и оценить разные способы представления множеств.

## 2. Задание

Необходимо найти множество, содержащее цифры, имеющиеся в A или B или являющиеся общими для C и D.

Формализация задания:  $E = A \cup B \cap (C \cap D)$

Необходимо решить задание четырьмя способами представления множеств:

- массив
- список
- машинное слово
- массив битов

В каждом случае оценить время, затрачиваемое на обработку множеств и сравнить по данному критерию все способы представления множеств.

### 3. Контрольные тесты

Тест 1.

All Output ↕

Clear

Операции над множествами (десятичные цифры)

Введите множества...

A=1234

B=56

C=7890

D=90

Введены множества: A=1234

B=56

C=7890

D=90

A|B|(C&D) (массивы): E=90123456

Время выполнения = 937 мкс

Представление

в виде списков : La=4321

Lb=65

Lc=0987

Ld=09

A|B|(C&D) (списки): Le=56123490

Время выполнения = 3573 мкс

Представление

вектором битов : bA=-----4321-

bB=---65-----

bC=987-----0

bD=9-----0

A|B|(C&D) (биты) : bE=9--6543210

Время выполнения = 216 мкс

Представление

машинным словом : wA=-----4321-

wB=---65-----

wC=987-----0

wD=9-----0

A|B|(C&D) (м. слово): wE=9--6543210

Время выполнения = 13 мкс

Тест 2.

```

All Output ↑ Clear

Операции над множествами (десятичные цифры)
Введите множества...
A=123456
B=56789
C=34560
D=67890

Введены множества: A=123456
                   B=56789
                   C=34560
                   D=67890
A|B|(C&D) (массивы): E=6012345789
Время выполнения = 2609 мкс

Представление
в виде списков : La=654321
                  Lb=98765
                  Lc=06543
                  Ld=09876
A|B|(C&D) (списки): Le=7891234560
Время выполнения = 6238 мкс

Представление
вектором битов : bA=---654321-
                  bB=98765-----
                  bC=---6543--0
                  bD=9876-----0
A|B|(C&D) (биты) : bE=9876543210
Время выполнения = 217 мкс

Представление
машинным словом : wA=---654321-
                  wB=98765-----
                  wC=---6543--0
                  wD=9876-----0
A|B|(C&D) (м.слово): wE=9876543210
Время выполнения = 13 мкс

```

#### 4. Временная сложность

Тип представления	Массивы символов	Список	Машинное слово	Вектор битов
Время обработки	$O(N^2)$	$O(N^2)$	$O(1)$	$O(1)$

#### 5. Результаты измерения времени обработки

Исходные данные	Время обработки, мкс.			
	Массивы символов	Список	Машинное слово	Вектор битов
A=[1234] B=[56] C=[7890] D=[90]	937	3573	13	216
A=[123456] B=[56789] C=[34560] D=[67890]	2609	6238	13	217

Как видно из данной таблицы, время обработки множеств, представленных массивами символов и списком, существенно изменяется при изменении размера исходных данных. При представлении множеств машинным словом и вектором битов изменения времени обработки не столь существенны.

## 6. Выводы

- 1) Под массивы выделяется память с запасом, имеется удобный доступ к элементам массива посредством индекса, обработка множеств представленных массивами имеет квадратичную временную сложность.
- 2) Список удобный способ представления множеств в памяти, память выделяется под каждый элемент множества и указатель на следующий элемент, однако обработка множеств списками самый долгий способ из представленных, работа производится с указателями, что требует большего времени. Имеет квадратичную временную сложность.
- 3) Представление множеств машинным словом является самым эффективным для выполнения данного задания. Операции над множествами в форме машинного слова выполняются за один шаг алгоритма независимо от мощности множеств, т. е. имеют временную сложность  $O(1)$ . Недостаток способа - отсутствие удобного доступа к каждому биту машинного слова, как к элементу массива.
- 4) Вектор битов имеет константную временную сложность, так как мощность универсума не изменяется.

## 7. Список использованных источников

1. Алгоритмы и структуры данных: методические указания к лабораторным работам, практическим занятиям и курсовому проектированию. Ч. 1. Вып. 1701 (для заочников) / сост.: П. Г. Колинько. — СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2017. — 64 с.
2. Курс видео уроков по C++. <https://www.youtube.com/playlist?list=PLbmlzoDQrXVFC13GjpPrJxI6mzTiX65gs>
3. Абстрактные типы данных. <http://algotlist.manual.ru/ds/basic/index.php>

## 8. Приложение

main.cpp - Файл содержит исходный код программы, в нем представлены способы представления и обработки множеств, расчет времени выполнения (массивы символов, списки, машинное слово, вектор битов).

```
// Задание выполнил студент группы 5005 Зинатуллин Азат.
// Вариант №21. Универсум десятичные числа.
// Множество, содержащее цифры, имеющиеся в A или B или являющиеся общими для C и D.
// Формализация задания: E=A|B|(C&D)

#include <iostream>
#include <cstring>
#include <time.h>

using namespace std;

struct EL {
    char d; //Элемент множества-списка
    EL * next;
    EL(char d, EL * next = nullptr) : d(d), next(next) { } //Конструктор
};

const int U = 10;

int main()
{
    setlocale(LC_ALL, "Russian");
    int nA, nB, nC, nD, nE=0, //Мощности множеств
    wA, wB, wC, wD, wE, //Память для машинных слов
    bA[U] = { 0 }, //Пустые массивы битов
    bB[U] = { 0 },
    bC[U] = { 0 },
    bD[U] = { 0 },
    bE[U]; //Память для результата
    EL * La, *Lb, *Lc, *Ld, *Le; //Указатели на списки
    char A[20], B[20], C[20], D[20], //Исходные множества (память с запасом на ввод)
    E[U + 1]; //Память под результат – массив

    cout << "\nОперации над множествами (десятичные цифры)";
```

```

cout << "\n Введите множества...\n";
cout << "A="; cin >> A; nA = strlen(A); //Ввод строки и подсчёт её длины
cout << "B="; cin >> B; nB = strlen(B); //(удобнее всего вводить строки)
cout << "C="; cin >> C; nC = strlen(C);
cout << "D="; cin >> D; nD = strlen(D);

/*
//Вариант: генерация исходных данных (проще всего генерировать м. слова)
srand(time(nullptr));
wA = rand( ); //Получение машинных слов
wB = rand( );
//Преобразование в массивы (строки символов)...
nA = 0;
for(int i = 0; i < U; ++i)
if(wA & (1<<i)) A[nA++] = i+'0';
A[nA] = '\0';
nB = 0;
for(int i = 0; i < U; ++i)
if(wB & (1<<i)) B[nB++] = i+'0';
B[nB] = '\0';
*/

//Преобразование строки символов в список
La = nullptr; //Список пуст
for (int i = 0; A[i]; ++i)
{
    EL *e = new EL(A[i], La); //Конструктор создаёт и сразу инициализирует
    La = e; //Вставка в список (в начало: порядок безразличен!)
}
Lb = nullptr;
for (int i = 0; B[i]; ++i)
{
    EL *e = new EL(B[i], Lb);
    Lb = e;
}
Lc = nullptr;
for (int i = 0; C[i]; ++i)
{
    EL *e = new EL(C[i], Lc);
    Lc = e;
}
Ld = nullptr;
for (int i = 0; D[i]; ++i)
{
    EL *e = new EL(D[i], Ld);
    Ld = e;
}

//Преобразование строк символов в массивы битов и машинные слова
wA = 0;
for (int i = 0; A[i]; ++i)
{
    bA[A[i] - '0'] = 1; //B [ ] – преобразование символа в номер бита
    wA |= (1 << (A[i] - '0')); //Здесь – сдвиг 1 на номер бита
}
wB = 0;
for (int i = 0; B[i]; ++i)
{
    bB[B[i] - '0'] = 1;
    wB |= (1 << (B[i] - '0'));
}
wC = 0;
for (int i = 0; C[i]; ++i)
{
    bC[C[i] - '0'] = 1;
    wC |= (1 << (C[i] - '0'));
}
wD = 0;
for (int i = 0; D[i]; ++i)
{
    bD[D[i] - '0'] = 1;
    wD |= (1 << (D[i] - '0'));
}

// Цикл на 100000 повторений для расчета времени выполнения операций с множествами,
// представленными в виде массива символов.
long time = 100000;
long start_time_array = clock(); // Подсчет тиков внутренних часов с помощью функции clock()
// до начала выполнения операций с множествами символов и
// запись результата в переменную.
for(int t = 0; t < time; ++t)
{

```

```

    nE=0;
    //Подсчет конечного множества через массивы символов
    for (int i = 0; C[i]; ++i) // пересечение множеств C и D.
    {
        bool f = true;
        for (int j = 0; D[j]&&f; ++j)
            if (C[i] == D[j])
                { E[nE]=C[i]; nE++; f=false; E[nE] = 0;}
    }

    for (int j = 0; A[j]; ++j) // добавляем недостающее из A
    {
        bool f = true;
        for (int i = 0; E[i]; ++i)
            if (E[i] == A[j]) f = false;
        if (f) {E[nE++] = A[j];E[nE] = 0; }
    }

    for (int j = 0; B[j]; ++j) //и добавляем недостающее из B
    {
        bool f = true;
        for (int i = 0; i<E[i]; ++i)
            if (E[i] == B[j]) f = false;
        if (f) {E[nE++] = B[j];E[nE] = 0; }
    }

    long end_time_array = clock(); // Запись в переменную после окончаний вычислений
    long time_array = (end_time_array - start_time_array); // Разница между начальным и конечным
                                                    //временем выполнения вычислений

    long start_time_bit = clock();
    for(long t = 0; t < time; ++t)
    {
        //Операции над массивами битов:
        for (int i = 0; i<U; ++i)
            bE[i] = (bC[i]&bD[i])|bA[i]|bB[i];
    }
    long end_time_bit = clock();
    long time_bit = (end_time_bit - start_time_bit);

    long start_time_spisok = clock();
    for(long t = 0; t < time; ++t)
    {
        //Операции над списками:
        Le = nullptr;
        for (EL* u = Lc; u; u = u->next)
        {
            bool f=true;
            for (EL* v = Ld; v&&f; v = v->next)
                if (u->d == v->d)
                {
                    EL* e = new EL(u->d, Le);
                    Le = e;
                    f=false;
                }
        }
        for (EL* u = La; u; u = u->next)
        {
            bool f = true;
            for (EL* v = Le; v; v = v->next)
                if (u->d == v->d) f=false;
            if(f)
            {
                EL* e = new EL(u->d, Le);
                Le = e;
            }
        }
        for (EL* u = Lb; u; u = u->next)
        {
            bool f = true;
            for (EL* v = Le; v; v = v->next)
                if (u->d == v->d) f=false;
            if(f)
            {
                EL* e = new EL(u->d, Le);
                Le = e;
            }
        }
    }

    long end_time_spisok = clock();
    long time_spisok = (end_time_spisok - start_time_spisok);

```

```

long start_time_mw = clock();
for(long t = 0; t < time; ++t)
{
    //Операции для машинных слов
    wE = (wC & wD) | wA | wB;
}

long end_time_mw = clock();
long time_mw = (end_time_mw - start_time_mw);

//===== ВЫВОД РЕЗУЛЬТАТОВ =====
cout << "\nВведены множества: A=" << A
<< "\n          B=" << B
<< "\n          C=" << C
<< "\n          D=" << D;
cout << "\nA|B|(C&D) (массивы): E=" << E ;
cout << "\nВремя выполнения = " << (time_array/50) << " мкс" << endl; // перевод времени в
микросекунды

cout << "\n\nПредставление ";
cout << "\n  в виде списков : La=";
for (EL * v = La; v; v = v->next) cout << v->d;
cout << "\n          Lb=";
for (EL * v = Lb; v; v = v->next) cout << v->d;
cout << "\n          Lc=";
for (EL * v = Lc; v; v = v->next) cout << v->d;
cout << "\n          Ld=";
for (EL * v = Ld; v; v = v->next) cout << v->d;
cout << "\nA|B|(C&D) (списки): Le=";
for (EL * v = Le; v; v = v->next) cout << v->d;
cout << "\nВремя выполнения = " << (time_spisok/50) << " мкс" << endl;

cout << "\n\nПредставление ";
cout << "\n вектором битов : bA=";
for (int i = U - 1; i >= 0; --i)
{
    if (bA[i]) cout << char(i + '0'); //Приведение к типу char здесь нужно.
    else cout << "-";
}
cout << "\n          bB=";
for (int i = U - 1; i >= 0; --i)
{
    if (bB[i]) cout << char(i + '0');
    else cout << "-";
}
cout << "\n          bC=";
for (int i = U - 1; i >= 0; --i)
{
    if (bC[i]) cout << char(i + '0');
    else cout << "-";
}
cout << "\n          bD=";
for (int i = U - 1; i >= 0; --i)
{
    if (bD[i]) cout << char(i + '0');
    else cout << "-";
}
cout << "\nA|B|(C&D) (биты) : bE=";
for (int i = U - 1; i >= 0; --i)
{
    if (bE[i]) cout << char(i + '0');
    else cout << "-";
}
cout << "\nВремя выполнения = " << (time_bit/50) << " мкс" << endl;

cout << "\n\nПредставление ";
cout << "\n машинным словом : wA=";
for (int i = U-1; i >=0 ; --i)
{
    if ((wA>>i)&1) cout << char(i + '0');
    else cout << "-";
}
cout << "\n          wB=";
for (int i = U-1; i >=0 ; --i)
{
    if ((wB>>i)&1) cout << char(i + '0');
    else cout << "-";
}
cout << "\n          wC=";
for (int i = U-1; i >=0 ; --i)
{
    if ((wC>>i)&1) cout << char(i + '0');

```

```

        else cout << "-";
    }
    cout << "\n      wD=";
    for (int i = U-1; i >=0 ; --i)
    {
        if ((wD>>i)&1) cout << char(i + '0');
        else cout << "-";
    }
    cout << "\nA|B|(C&D)(м.слово):wE=";
    for (int i = U-1; i >=0 ; --i)
    {
        if ((wE>>i)&1) cout << char(i + '0');
        else cout << "-";
    }
    cout << "\nВремя выполнения = " << (time_mw/50) << " мкс" << endl;
    return 0;
}

```