

Отчет по Лабораторной работе №3 А.Труфанова СКБ221

Алгоритм решения задачи и обработка флагов

Основной алгоритм и обработку флагов содержит в себе файл main.cpp

Флаги --tofile и --fromfile указывают на параметры ввода и вывода. Для их обработки я добавила в функцию main параметры argc и **argv. Сравнением строк определяется положение флагов. В случае, когда за флагом идет название файла, оно запоминается в переменную char*. Учтена возможность введения двух флагов одновременно, а также некорректный ввод флагов.

Алгоритм одинаков для введения флагов по отдельности и одновременно. Для решения поставленной задачи я создала двумерный массив **a**, где хранятся координаты отрезков. Двумерный массив нужен, чтобы идентифицировать начало (1) и конец (-1). Далее массив сортируется методом пузырька. Для нахождения объединения понадобился счетчик bal, который увеличивается, если элемент отсортированного массива является началом отрезка и уменьшается, если встретился конец отрезка. Таким образом, в момент обнуления счетчика мы получаем координату конца объединения. Когда счетчик равен единице, элемент является началом отрезка. Нам нужно сохранить минимальную координату. Для этого я написала функцию min, которая содержится в файле solution.cpp.

Makefile

В нем собраны файлы cpp в объектные файлы и в последствии в исполняемый файл. Clean для быстрой очистки всех результатов сборки проекта. Distclean для удаления файлов txt.

Вывод

Я научилась составлять makefile и работать с GitHub

Листинг 1: Основной алгоритм и обработка флага

```
#include<iostream>
#include<fstream>
#include<cstring>
#include"solution.h"
int main(int argc, char **argv){
    FILE *fp;
    bool tofile=false;
    bool fromfile=false;
    char *name1;
    char *name2;
    bool testfile=false;
    bool tofilename=false;
    if(argc==1){
        std::cerr<<"Nothing was read\n";
        return EXIT_FAILURE;
    }
    if(argc>5){
        std::cerr<<"Too many flags\n";
        return EXIT_FAILURE;
    }
    else{
        for (int i=1;i<argc;++i){
            if(std::strcmp(argv[i],"--tofile")==0){
                tofile=true;
                if(!argv[i+1]) (std::strcmp(argv[i],"--fromfile")==0){
                    std::ofstream oFile("test.txt");
                }
            }
        }
    }
}
```

```

        testfile=true;
    }
    else{
        name1=argv[i+1];
    }
}
if(std::strcmp(argv[i], "--fromfile")==0){
    fromfile=true;
    if((!argv[i+1]) (std::strcmp(argv[i], "--tofile")==0){
        std::cerr<<"File name not specified\n";
        return EXIT_FAILURE;
    }
    else{
        name2=argv[i+1];
    }
}

}

if(!fromfile&&!tofile){
    std::cerr<<"Incorrect input\n";
    return EXIT_FAILURE;
}
if (fromfile){
    std::fstream input(name2);
    fp=fopen(name2, "r");
    double segment;
    int n;
    input>>n;
    double a[2*n][2];
    for(int i=0; i<2*n; i++){
        input>>segment;
        if(i%2==0){
            a[i][0]=segment;
            a[i][1]=1;
        }
        else{
            a[i][0]=segment;
            a[i][1]=-1;
        }
    }
    for(int i=0; i<2*n; i++){
        if((i%2!=0)&&(a[i-1][0]>a[i][0])){
            std::cerr<<"Nothing found because of incorrect
input"<<std::endl;
            return EXIT_FAILURE;
        }
    }
    fclose(fp);
    double temp;
    int t;
    for (int i = 0; i < 2*n - 1; i++) {
        for (int j = 0; j < 2*n - i - 1; j++) {
            if (a[j][0] > a[j + 1][0]) {
                temp = a[j][0];
                t=a[j][1];
                a[j][0] = a[j + 1][0];
                a[j][1]=a[j+1][1];
                a[j + 1][0] = temp;
                a[j+1][1]=t;
            }
        }
    }
}

```

```

    }
}
}
if(!tofile){
    int bal=0;
    double x=1000000000;
    for (int i=0;i<2*n;i++){
        if (a[i][1]==1){
            bal++;
        }
        if (a[i][1]==-1){
            bal--;
        }
        if (bal==1){
            x=min(x,a[i][0]);
        }
        if (bal==0){
            std::cout<<x<<" "<<a[i][0]<<std::endl;
            x=1000000000;
        }
    }
}
else if(tofile=true){
    if(testfile){
        std::ofstream out;
        out.open("test.txt");
        if(out.is_open()){
            int bal=0;
            double x=1000000000;
            for (int i=0;i<2*n;i++){
                if (a[i][1]==1){
                    bal++;
                }
                if (a[i][1]==-1){
                    bal--;
                }
                if (bal==1){
                    x=min(x,a[i][0]);
                }
                if (bal==0){
out<<x<<" "<<a[i][0]<<std::endl;
                    x=1000000000;
                }
            }
        }
        out.close();
        std::cout<<"Answer in file: test.txt"<<std::endl;;
    }
    else{
        std::fstream output(nam1);
        fp=fopen(nam1,"w");
        if(output.is_open()){
            int bal=0;
            double x=1000000000;
            for (int i=0;i<2*n;i++){
                if (a[i][1]==1){
                    bal++;
                }
                if (a[i][1]==-1){
                    bal--;
                }
            }
        }
    }
}

```

```

        }
        if (bal==1){
            x=min(x,a[i][0]);
        }
        if (bal==0){
            output<<x<<" "<<a[i][0]<<std::endl;
            x=1000000000;
        }
    }

    }

    output.close();
    std::cout<<"Answer in file: "<<name1<<std::endl;;

}

}

}

if(tofile&&!fromfile){
    int n;
    std::cin>>n;
    double segment;
    double a[2*n][2];
    for(int i=0;i<2*n; i++){
        std::cin>>segment;
        if(i%2==0){
            a[i][0]=segment;
            a[i][1]=1;
        }
        else{
            a[i][0]=segment;
            a[i][1]=-1;
        }
    }

    for(int i=0;i<2*n;i++){
        if((i%2!=0)&&(a[i-1][0]>a[i][0])){
            std::cerr<<"Nothing found because of incorrect
input"<<std::endl;

            return EXIT_FAILURE;
        }
    }

    double temp;
    int t;
    for (int i = 0; i < 2*n - 1; i++) {
        for (int j = 0; j < 2*n - i - 1; j++) {
            if (a[j][0] > a[j + 1][0]) {
                temp = a[j][0];
                t=a[j][1];
                a[j][0] = a[j + 1][0];
                a[j][1]=a[j+1][1];
                a[j + 1][0] = temp;
                a[j+1][1]=t;
            }
        }
    }

    if(!tofile){
        int bal=0;
        double x=1000000000;
        for (int i=0;i<2*n;i++){
            if (a[i][1]==1){

```

```

        bal++;
    }
    if (a[i][1]==-1){
        bal--;
    }
    if (bal==1){
        x=min(x,a[i][0]);
    }
    if (bal==0){
        std::cout<<x<<" "<<a[i][0]<<std::endl;
        x=1000000000;
    }
}
}
else if(tofile=true){
    if(testfile){
        std::ofstream out;
        out.open("test.txt");
        if(out.is_open()){
            int bal=0;
            double x=1000000000;
            for (int i=0;i<2*n;i++){
                if (a[i][1]==1){
                    bal++;
                }
                if (a[i][1]==-1){
                    bal--;
                }
                if (bal==1){
                    x=min(x,a[i][0]);
                }
                if (bal==0){
                    out<<x<<" "<<a[i][0]<<std::endl;
                    x=1000000000;
                }
            }
        }
    }
}

```

настя камри 3.5, [28.11.2022 19:47]

```

    out.close();
    std::cout<<"Answer in file: test.txt"<<std::endl;;
}
else{
    std::fstream output(namel);
    fp=fopen(namel,"w");
    if(output.is_open()){
        int bal=0;
        double x=1000000000;
        for (int i=0;i<2*n;i++){
            if (a[i][1]==1){
                bal++;
            }
            if (a[i][1]==-1){
                bal--;
            }
            if (bal==1){
                x=min(x,a[i][0]);
            }
            if (bal==0){
                output<<x<<" "<<a[i][0]<<std::endl;
                x=1000000000;
            }
        }
    }
}

```

```

        }
    }

    }
    output.close();
    std::cout<<"Answer in file: "<<name1<<std::endl;;

    }

    }

    }

    }
    return 0;
}

```

Листинг 2: Функция min

```

double min(double a, double b){
    if(a<b){
        return a;
    }
    else{
        return b;
    }
}

```