Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут ім. Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

ЗВІТ

про виконання лабораторного практикуму №6(2)

Виконав:

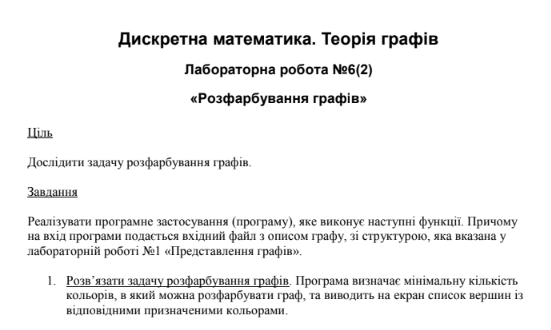
студент 1- го курсу ФІОТ

групи *ІП-91*

*Кінчур Вадим Вікторович*

Київ 2020

1. Умова лабораторної роботи



1. Програмний код (С++):

#include <iostream>

#include <fstream>

#include <list>

#include <vector>

#include <algorithm>

using namespace std;

void get\_input(const char\*, list<int>\*&, int&);

template<typename T>

T\* fill\_array(const T, const int);

int get\_degree(int\*, const int);

int\* get\_degrees(list<int>\*, const int);

vector<list<int>> get\_colors(list<int>\*, const int);

bool is\_correct\_color(list<int>\*, int\*, const int, const int, const int);

int\* get\_vertex\_order(int\*, const int);

void print\_colors(list<int>\*, const int);

int main() {

int v\_num;

list<int>\* adj\_list = nullptr;

get\_input("input.txt", adj\_list, v\_num);

print\_colors(adj\_list, v\_num);

delete[] adj\_list;

}

template<typename T>

T\* fill\_array(const T filler, const int size) {

T\* arr = new T[size];

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

arr[i] = filler;

return arr;

}

void get\_input(const char\* file\_name, list<int>\*& adj\_list, int& v\_num) {

ifstream file\_handler(file\_name);

int e\_num;

file\_handler >> v\_num >> e\_num;

adj\_list = new list<int>[v\_num];

for (auto i = 0; i < e\_num; ++i) {

int start, end;

file\_handler >> start >> end;

adj\_list[start - 1].push\_back(end - 1);

adj\_list[end - 1].push\_back(start - 1);

}

file\_handler.close();

}

bool is\_correct\_color(list<int>\* adj\_l, int\* colors, const int v, const int color, const int v\_num) {

for (auto vertex : adj\_l[v]) {

if (colors[vertex] == color)

return false;

}

return true;

}

int\* get\_degrees(list<int>\* adj\_l, const int v\_num) {

int\* degrees = new int[v\_num];

for (auto v = 0; v < v\_num; ++v) {

degrees[v] = adj\_l[v].size();

}

return degrees;

}

int get\_degree(int\* deg, const int v\_num) {

return \*max\_element(deg, deg + v\_num) + 1;

}

int\* get\_vertex\_order(int\* degrees, const int v\_num) {

int\* order = new int[v\_num];

for (auto i = 0; i < v\_num; ++i)

order[i] = i;

for (auto i = 0; i < v\_num; ++i) {

for (auto j = v\_num - 1; j > i; --j) {

if (degrees[j] > degrees[j - 1]) {

swap(degrees[j], degrees[j - 1]);

swap(order[j], order[j - 1]);

}

}

}

return order;

}

vector<list<int>> get\_colors(list<int>\* adj\_l, const int v\_num) {

vector<list<int> > color\_set;

auto\* degrees = get\_degrees(adj\_l, v\_num);

const auto max\_colors = get\_degree(degrees, v\_num);

auto\* colors = fill\_array(-1, v\_num);

auto\* order = get\_vertex\_order(degrees, v\_num);

for (auto i = 0; i < v\_num; ++i) {

for (auto color = 0; color < max\_colors; ++color) {

if (is\_correct\_color(adj\_l, colors, order[i], color, v\_num)) {

if (color >= color\_set.size())

color\_set.push\_back(list<int>{});

colors[order[i]] = color;

color\_set[color].push\_back(order[i]);

break;

}

}

}

delete[] degrees;

delete[] colors;

delete[] order;

return color\_set;

}

void print\_colors(list<int>\* adj\_l, const int v\_num) {

auto color\_set = get\_colors(adj\_l, v\_num);

for (auto i = 0; i < color\_set.size(); ++i) {

cout << "Color " << i + 1 << '\n';

for (auto el : color\_set[i]) {

cout << el + 1 << " ";

}

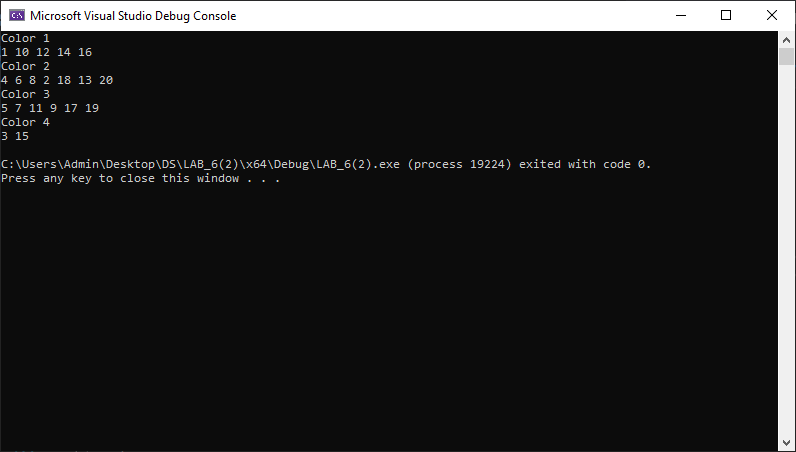
cout << '\n';

}

}

1. Результат виконання

**Вхідні дані:**

20 43

1 2

1 4

1 5

1 6

1 7

1 8

2 3

2 9

2 10

3 4

3 10

3 11

4 5

4 11

4 12

5 6

5 13

5 14

6 7

6 14

6 15

7 8

7 15

7 16

8 9

8 16

8 17

9 10

9 18

10 11

10 18

10 19

11 12

11 20

12 13

12 20

13 14

14 15

15 16

16 17

17 18

18 19

19 20