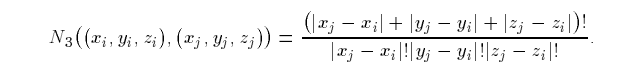
Для решения данной задачи используется математическая модель решетчатого графа, в котором минимальный путь между двумя вершинами вычисляется по формуле:



Для подсчета количества кратчайших путей используется формула:



Исходный код программы с комментариями:

#include <iostream>

#include <ctime>

#include <vector>

#include <fstream>

#include <windows.h>

#include <ctime>

#include <math.h>

using namespace std;

vector<int> coordStart,coordFinish;

//функция вычисления факториала

unsigned long fact(unsigned long num)

{

if(num<=1) return 1;

else

return num\*fact(num-1);

}

//обработка данных

void analyze()

{

//числитель целевой дроби

unsigned long chisl = abs(coordStart[0]-coordFinish[0]) + abs(coordStart[1]-coordFinish[1]) + abs(coordStart[2]-coordFinish[2]);

//вычисляем факториал

chisl = fact(chisl);

//вычисляем факториалы составляющих знаменателя

unsigned long zn1 = fact(abs(coordStart[0]-coordFinish[0]));

unsigned long zn2 = fact(abs(coordStart[1]-coordFinish[1]));

unsigned long zn3 = fact(abs(coordStart[2]-coordFinish[2]));

unsigned long znam = zn1\*zn2\*zn3;

//целевая дробь

unsigned long count = chisl/znam;

if(count>1000000) count = count % 109;

cout << "Count is " << count << endl;

}

bool rfile(const string& name)

{

ifstream file;

file.open(name);

if(file.is\_open()) {

file >> coordStart[0] >> coordStart[1] >> coordStart[2] ;

file >> coordFinish[0] >> coordFinish[1] >> coordFinish[2] ;

for(int i=0;i<3;i++)

{

int j = max(coordFinish[i],coordStart[i]);

if(j>1000)

{

cerr << "Very high input data" << endl;

return 0;

}

}

file.close ();

cout << coordStart[0] << coordStart[1] << coordStart[2] << endl;

cout << coordFinish[0] << coordFinish[1] << coordFinish[2] << endl;

return 1;

}

else

{

cerr << "File not found" << endl;

return 0;

}

}

int main(int argc, char \*argv[])

{

coordStart.reserve(3);

coordFinish.reserve(3);

//время выполнения

int start = clock();

//чтение из файла

if(!rfile("input.txt")) return 0;

//обработка данных

analyze();

//останов часов

int stop = clock();

cout << ((float)(stop-start)) / CLOCKS\_PER\_SEC << " seconds" << endl;

return 0;

}