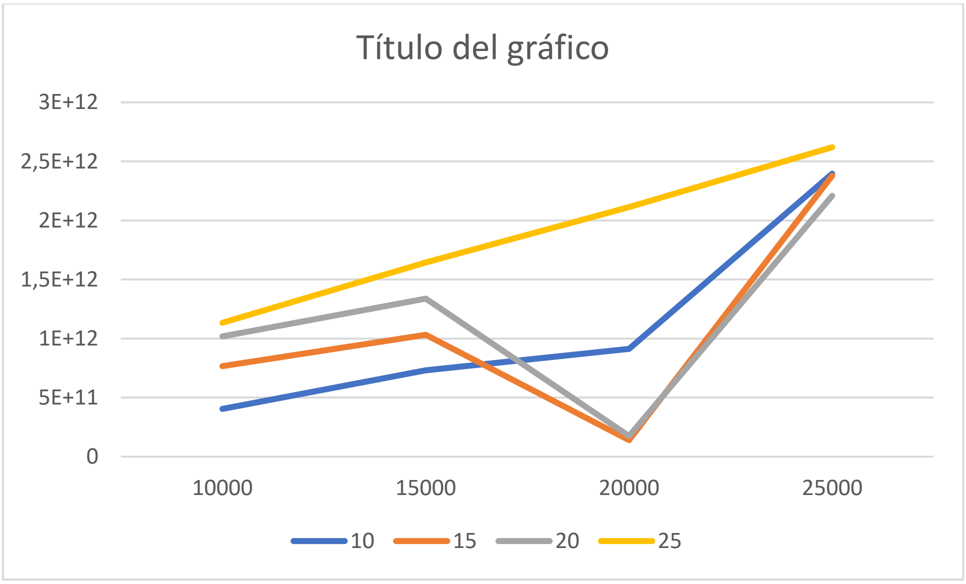
	10	15	20	25
10000	4,04533E+11	7,6651E+11	1,01807E+12	1,13415E+12
15000	7,30135E+11	1,0312E+12	1,33733E+12	1,64498E+12
20000	9,13065E+11	1,4035E+11	1,74661E+11	2,11265E+12
25000	2,39648E+12	2,3793E+12	2,20908E+12	2,62034E+12



```
#include <utility>

#include <random>

#include <iostream>

#include <chrono>

#include <vector>

#include <cmath>

#include <math.h>

using namespace std;


double random() {

   random_device rd;

   mt19937 gen(rd());
```

```
    uniform_real_distribution<> dis(1.0, 2.0);

    return dis(gen);

}


double diferencia_eucladiana(vector<double>vector_1, vector<double>vector_2, double lado)
{

    double temp = 0, temp1;

    for (double i = 0; i < lado; i++) {

        temp1 = vector_1[i] - vector_2[i];

        temp1 *= temp1;

        temp += temp1;

    }

    temp = sqrt(temp);

    return temp;

}


void imprimir(vector<vector<double> > myvector, double LENGTH, double WIDTH) {

    for (double i = 0; i < LENGTH; i++) {

        for (double j = 0; j < WIDTH; j++) {

            cout << myvector[i][j] << " ";

        }

        cout << endl;

    }

}


void rellenar(vector<vector<double> >& todo, double abajo, double lado) {

    for (double i = 0; i < 10; i++) {


        //cout << "rellenado " << i << endl;

        for (double j = 0; j < lado; j++) {

            todo[i][j] = random();
        }
    }

    for (double i = 10, j = 0; i < abajo && j <= 100; i++, j++) {
        //cout << "rellenado " << i << endl;
        todo[i] = todo[j];
        if (j + 1 > 10)
            j = 0;
    }
}

void distancia(vector<vector<double> >& todo, double abajo, double lado) {
    double dis;
    chrono::time_point<std::chrono::high_resolution_clock> start, end1, end2, end3;

    start = std::chrono::high_resolution_clock::now();

    int64_t duration1=0, duration2, duration3;
    for (int indice = 0; indice < abajo - 1; indice++) {
        cout << indice << endl;
        for (int i = indice + 1; i < abajo; i++) {
            dis = diferencia_eucladiana(todo[indice], todo[i], lado);
            //cout << "la distancia entre el vector " << indice << " y el " << i << " es " << dis << endl;
        }

        if (indice == 10000) {
            end1 = chrono::high_resolution_clock::now();
             duration1 = std::chrono::duration_cast<std::chrono::nanoseconds>(end1 -
start).count();

        }
        if (indice == 15000) {
            end2 = chrono::high_resolution_clock::now();
             duration2 = std::chrono::duration_cast<std::chrono::nanoseconds>(end2 -
start).count();
        }

        if (indice == 20000) {
            end3 = chrono::high_resolution_clock::now();
             duration3 = std::chrono::duration_cast<std::chrono::nanoseconds>(end3 -
start).count();
        }

    }

    cout << duration1 << endl << "termino" << endl;
    cout << duration2 << endl << "termino" << endl;
    cout << duration3 << endl << "termino" << endl;
}

int main() {
    double abajo = 20000;
    double lado = 15;
    vector<vector<double> > todo(abajo, vector<double>(lado));
    chrono::time_point<std::chrono::high_resolution_clock> start, end, end2, end3;

    start = std::chrono::high_resolution_clock::now();
    rellenar(todo, abajo, lado);
    cout << "todos los vectores rellenados" << endl;
    //imprimir(todo, abajo, lado);
    cout << "vectores" << abajo << " x " << lado << endl;
    distancia(todo, abajo, lado);
    end = chrono::high_resolution_clock::now();
    int64_t duration4 = std::chrono::duration_cast<std::chrono::nanoseconds>(end - start).count();
    cout << duration4 << endl << "termino" << endl;


}
```