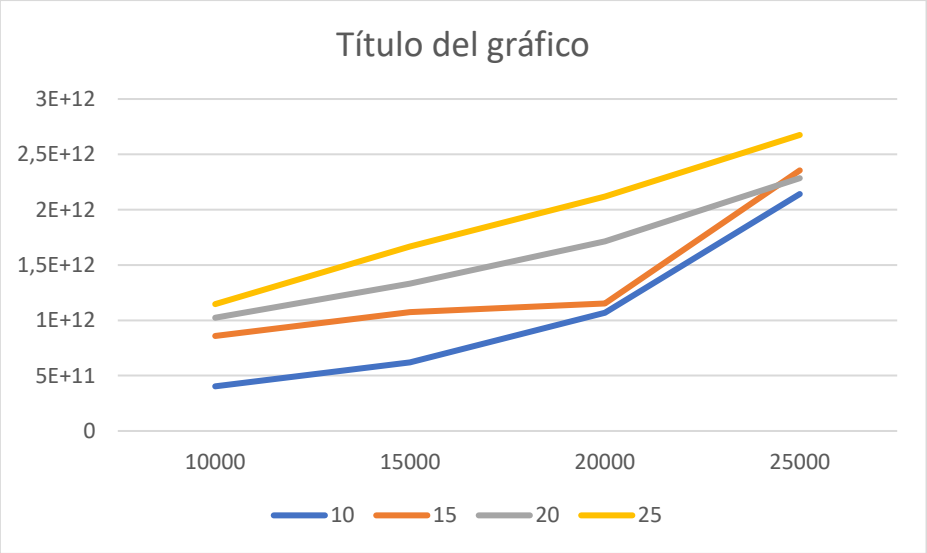
| | 10 | 15 | 20 | 25 |
|---|---|---|---|---|
| **10000** | 4,03579E+11 | 8,58627E+11 | 1,02359E+12 | 1,14652E+12 |
| **15000** | 6,21268E+11 | 1,07458E+12 | 1,33152E+12 | 1,66613E+12 |
| **20000** | 1,06854E+12 | 1,15236E+12 | 1,71306E+12 | 2,11798E+12 |
| **25000** | 2,14153E+12 | 2,35515E+12 | 2,28465E+12 | 2,67531E+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>v1, vector<double>v2, double
elementos) {
    double suma = 0, resultado;
    for (double i = 0; i < lado; i++) {
        resultado = v1[i] - v2[i];
        resultado *= resultado;
        suma += resultado;
    }
    suma = sqrt(suma);
    return suma;
```

```
}


void poner_random(vector<vector<double> >& matriz, double arrays, double
elementos) {
    for (double i = 0; i < lado; i++) {
        for (double j = 0; j < lado; j++) {
            todo[i][j] = random();
        }
    }
}

void dis(vector<vector<double> >& matriz, double arrays, double elementos) {
    double resta;

    for (int i = 0; i < abajo - 1; i++) {
        cout << indice << endl;
        for (int j = i + 1; j < abajo; j++) {
            resta = diferencia_eucladiana(todo[i], todo[j], lado);
        }
    }
}

int main() {
    double arrays = 20000;
    double elementos = 15;
    vector<vector<double> > matriz(arrays, vector<double>(elementos));
    chrono::time_point<std::chrono::high_resolution_clock> start, end;

    start = std::chrono::high_resolution_clock::now();
    poner_random(matriz, arrays, elementos);
    dis(matriz, arrays, elementos);
    end = chrono::high_resolution_clock::now();
    int64_t duration4 = std::chrono::duration_cast<std::chrono::nanoseconds>(end
- start).count();
    cout << duration4 << endl << "termino" << endl;

}
```