Факультет программной инженерии и компьютерной техники

Алгоритмы и структуры данных

Введение в алгоритмы

Преподаватели: Косяков Михаил Сергеевич, Тараканов Денис Сергеевич

Выполнил: Кульбако Артемий Юрьевич

Р3212

# 1726. Кто ходит в гости…

#include <iostream>

using namespace std;

void quickSort(int array[], int left, int right) {

    int i = left, j = right;

    int pivot = array[(left + right) / 2];

    while(i <= j) {

        while (array[i] < pivot) i++;

        while (array[j] > pivot) j--;

        if (i > j) break;

        int t = array[i];

        array[i] = array[j];

        array[j] = t;

        i++;

        j--;

    }

    if (left < j) quickSort(array, left, j);

    if (i < right) quickSort(array, i, right);

}

int main() {

    long long n, sum = 0;

    cin >> n;

    int xCoords[n];

    int yCoords[n];

    for (long long i = 0; i < n; i++) cin >> xCoords[i] >> yCoords[i];

    quickSort(xCoords, 0, n - 1);

    quickSort(yCoords, 0, n - 1);

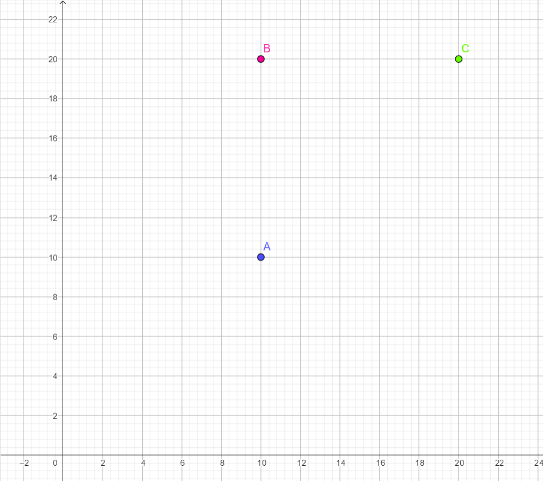
    for (long long i = 1; i < n; i++) sum += ((xCoords[i] - xCoords[i-1]) + (yCoords[i] - yCoords[i-1])) \* (i) \* (n - i) \* 2;

    sum = sum / (n \* (n - 1));

    cout << sum << endl;

    return 0;

}

Построим карту происходящего, для лучшего понимания. Так как по условию задачи все члены программного комитета ходят в гости ко всем одинаково, то каждый дом нужно связать с каждым, исключительно прямыми линиями (можно перемещаться только параллельно OX и OY). Построим на ней вектора возможных перемещений между домами. Чтобы найти среднее растояние, которое проходит член программного комитета от своего дома до дома своего товарища, необходимо найти расстояние от его дома, до каждого из нужных домов, и повторить это с каждым членом комитета.

|  |  |  |
| --- | --- | --- |
|  |  |  |

На этом этапе возникают две проблемы: 1-я - необходимость просчитывать заново длину дорожек, по которым уже кто-то ходил, и 2-я - как дать программе понять, что вектор между дорожками правильно направлен, чтобы не получать отрицательные значения длин векторов? Обе проблемы решаются сортировкой координат по возрастанию. Для этого реализуем алгоритм сортировки Хорна известный своим временем работы:

1. Выберем из массива опорный элемент - он будет равняться серединному элементу (pivot = array[(left + right) / 2]).
2. Разбиение: перераспределение элементов в массиве таким образом, что элементы меньше опорного помещаются перед ним, а больше или равные после.
3. Рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором меньше двух элементов.

Решение 1-ой: Количество координат каждого типа равняется количеству домов на карте. Значит в отсортированном массиве для каждого элемента будет

известна длина дорожки от i до (n - i) элемента (по которой уже ходил кто-то из членом клуба). Значит по каждой дороге пройдут (длинаПути \* i \* (n - i)) условных единиц расстояния.

Решение 2-ой: В отсортированном массиве мы всегда будем отнимать от большего x (или y) меньший (x или y), таким образом, не будем получать отрицательные значения (альтернативный вариант - функция abs(x1 - x2)).

Спроецируем карту города на оси OX и OY и посчитаем, суммарную длину всех путей. Чтобы найти средний путь члена комитета, поделим полученное число на количество существующих дорожек.