

LAPORAN PRAKTIKUM

DIVIDE AND CONQUER

Nama/ NIM : Whisnumurty Galih Ananta/ 1203210002
Mata Kuliah/ Kelas : Perancangan dan Analisis Algoritma/ IF-01-03
Dosen Pengampu :

Source Code : [praktikum-PAA/Praktikum 6 - Divide and Conquer/clopair.cpp at master · whisnumurtyga/praktikum-PAA · GitHub](https://github.com/whisnumurtyga/praktikum-PAA/tree/master/praktikum-PAA/Praktikum%206%20-%20Divide%20and%20Conquer/clopair.cpp)
Soal : [SPOJ.com - Problem CLOPPAIR](https://www.spoj.com/problems/CLOPPAIR/)

Penjelasan Soal

Dalam permasalahan ini, diberikan N titik pada bidang datar, dan tugas adalah mencari pasangan titik dengan jarak Euclidean terkecil di antara mereka. Jarak Euclidean antara dua titik (x_1, y_1) dan (x_2, y_2) . dijamin bahwa semua titik unik, dan hanya ada satu pasangan titik dengan jarak terkecil.

Anda diminta untuk membaca input yang terdiri dari N , yang merupakan jumlah titik, diikuti oleh N baris berisi koordinat X dan Y dari masing-masing titik. Setelah itu, Anda harus mencetak 3 angka a, b, c sebagai output, di mana a dan b (dengan $a < b$) adalah indeks (berbasis 0) dari pasangan titik dalam input, dan c adalah jarak di antara kedua titik tersebut, dibulatkan menjadi 6 digit desimal.

Anda harus menyelesaikan permasalahan ini dengan menggunakan algoritma "Closest Pair" yang menggunakan pendekatan divide and conquer. Algoritma ini membagi set titik menjadi bagian-bagian lebih kecil, menyelesaikan submasalah pada setiap bagian, dan kemudian menggabungkan solusi-solusi tersebut untuk mendapatkan solusi keseluruhan.

Contoh Input Output

Input:

```
5
0 0
-4 1
-7 -2
4 5
1 1
```

Output:

```
0 4 1.414214
```

Dalam contoh ini, kita diberikan 5 titik pada bidang dengan koordinat yang berbeda-beda. Tugas kita adalah mencari pasangan titik dengan jarak Euclidean terkecil di antara mereka.

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Posisi A	Posisi B	Value A	Value B	Euclidean Distance
0	1	(0,0)	(-4,1)	4.123106
0	2	(0,0)	(-7,2)	7.280110
0	3	(0,0)	(4,5)	6.403124
0	4	(0,0)	(1,1)	1.414214
1	2	(-4,1)	(-7,2)	4.242641
1	3	(-4,1)	(4,5)	8.944272
1	4	(-4,1)	(1,1)	5
2	3	(-7,2)	(4,5)	13.038405
2	4	(-7,2)	(1,1)	8.544004
3	4	(4,5)	(1,1)	5

Dalam output, kita mencantumkan indeks pasangan titik tersebut, di mana titik 0 adalah titik pertama (0, 0) dan titik 4 adalah titik kedua (1, 1). Jarak Euclidean di antara mereka adalah 1.414214, yang diambil dengan pembulatan 6 angka desimal. Jadi, outputnya adalah "0 4 1.414214".

Penjelasan Program

1. **Mendefinisikan tipe data `points`** yang merepresentasikan titik dengan atribut `x` (koordinat X), `y` (koordinat Y), dan `pos` (posisi/indeks titik dalam array).
2. **Mendefinisikan fungsi `xcompare` dan `ycompare`** yang digunakan untuk membandingkan dua titik berdasarkan koordinat X dan Y mereka.
3. **Mendefinisikan fungsi `dis`** yang menghitung jarak Euclidean antara dua titik menggunakan rumus matematika yang sesuai.
4. **Mendefinisikan fungsi `brute`** yang menggunakan pendekatan brute force untuk mencari pasangan titik terdekat di dalam suatu range. Fungsi ini membandingkan jarak setiap pasangan titik dan mengupdate `bestmin`, `u`, dan `v` jika ditemukan jarak yang lebih kecil.
5. **Mendefinisikan fungsi `findclose`** yang digunakan untuk mencari pasangan titik terdekat dalam suatu range dengan menggunakan pendekatan divide and conquer. Fungsi ini mengurutkan titik-titik dalam range berdasarkan koordinat Y dan membandingkan jarak setiap pasangan titik dalam range tersebut. Jika ditemukan jarak yang lebih kecil, maka `bestmin`, `u`, dan `v` akan diupdate.
6. **Mendefinisikan fungsi `closepoints`** yang merupakan implementasi algoritma "Closest Pair" secara rekursif. Fungsi ini menerima array `a`, indeks awal (`s`), indeks akhir (`e`), dan ukuran array (`size`) sebagai parameter. Jika jumlah titik dalam range tersebut kurang dari atau sama dengan 3, maka fungsi `brute` dipanggil untuk mencari pasangan titik terdekat di range tersebut. Jika lebih dari 3, fungsi `closepoints` akan membagi range menjadi dua bagian dan mencari pasangan titik terdekat di kedua bagian tersebut. Kemudian, hasil dari kedua bagian tersebut akan digabungkan dengan mencari pasangan titik terdekat yang melintasi garis pembagi antara kedua bagian tersebut.
7. **Di dalam fungsi `main`**, dibaca input berupa jumlah titik (`n`) dan koordinat titik dari input. Setiap titik disimpan dalam array `a` dan atribut `pos` diisi dengan indeks titik tersebut. Array `a` diurutkan berdasarkan koordinat X menggunakan fungsi `xcompare`. Kemudian, fungsi `closepoints` dipanggil untuk mencari pasangan titik terdekat secara rekursif. Terakhir, hasilnya (indeks pasangan titik terdekat dan jaraknya) dicetak ke layar.

Dengan demikian, kode tersebut menggunakan algoritma "Closest Pair" untuk mencari pasangan titik terdekat dengan kompleksitas waktu $O(n \log n)$, di mana n adalah jumlah titik.