Closest Pair Report

Andreas Bitzilis, Christos Grigoriou and Dimos Zikos

September 12, 2016

Results

Our implementation produces the expected results on all input-output file pairs.

The following table shows the closest pairs in the input files wc-instance-14.txt. Here n denotes the number of points in the input, and (u, v) denotes a closest pair of points at distance d.

|  |  |  |  |
| --- | --- | --- | --- |
| n | u | v | d |
| 14 | (-0.5, 0.0) | (-0.125, 3.0) | 3.0234 |
|  |  |  |  |

Implementation details

We resort by y-coordinates in each recursive step. For the comparison of points close to s in Sy we inspect 15 points, as explained (5.10) of Kleinberg and Tardos, Algorithm Design, Addison–Wesley 2008. Here is the corresponding part of our code:

min = Double.MAX\_VALUE;

count ;

for (Point s : Sy)

count = 0;

for [int I = 0 ; I < Sy.length; i++]

if(count++ > 16)

break;

if (s.distance(Sy[i]) < min)

min = s.distance(S[i]);

We combine the information from the recursive calls in linearithmic time instead of linear, thus the relation has the form of , T(N) = 2T(N/2) + cNlogN. By unrolling the relation we identify that T(n) → cn Σlog2 n-1 log(n/2j). It lies between nlogn and n2.. We also took the chance to implement concepts introduced in Concurrent and Parallel Programming.