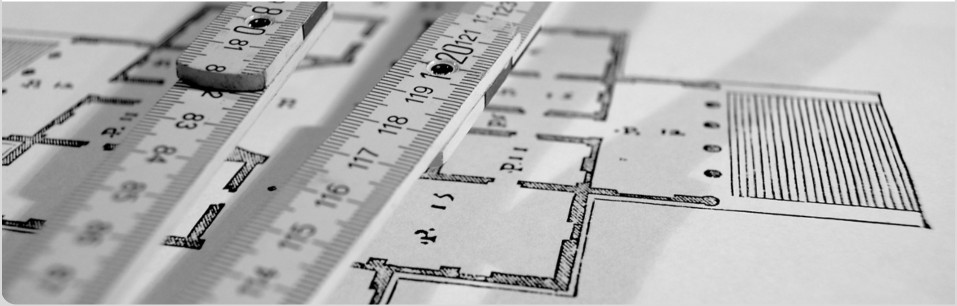
Karlsruhe Institute of Technology

Parallel Algorithm for Closest Pair Problem

Ge Wu | July 22, 2013

INSTITUTE FOR THEORETICAL INFORMATICS



Problem Description

Closest Pair Problem

- Given n **different unordered** points $P = \{p_1, p_2, ..., p_n\}$ in **unit square**:

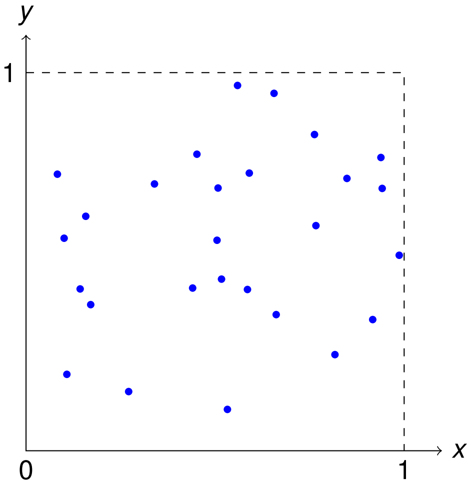
$$p_i = (x_i, y_i) \in (0,1) \times (0,1) \subset \mathbb{R}^2$$

- Find a pair of points with the smallest distance between them.
- Find any pair if there's a tie.

Background

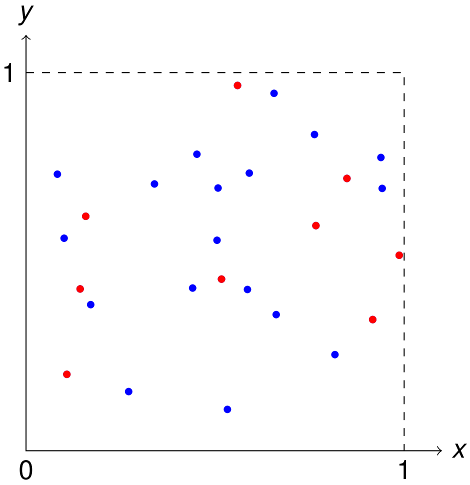
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



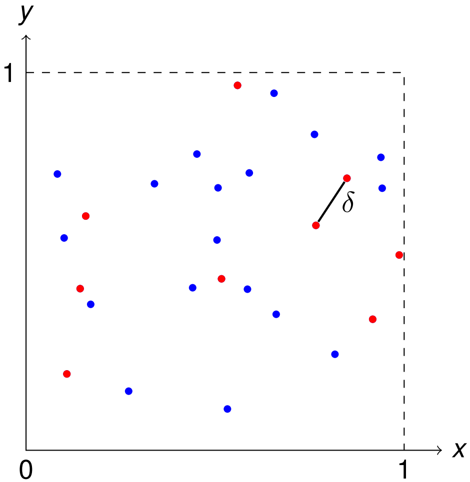
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



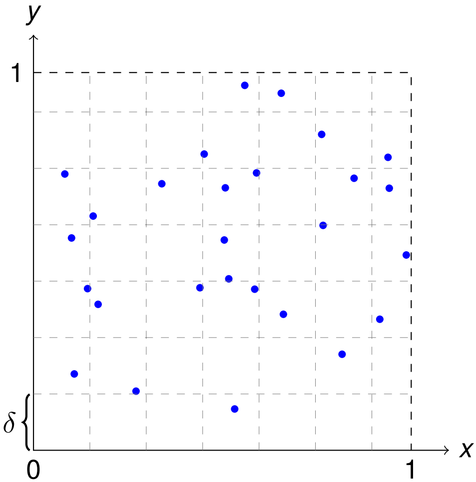
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



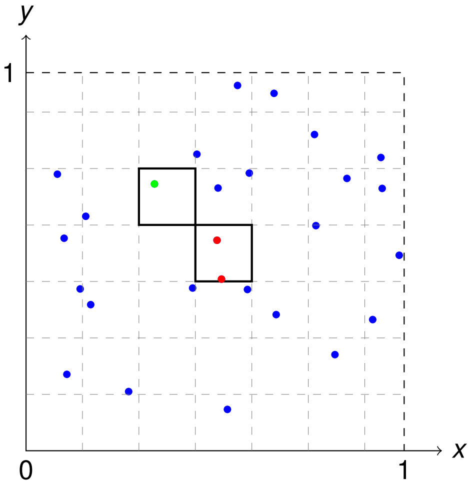
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



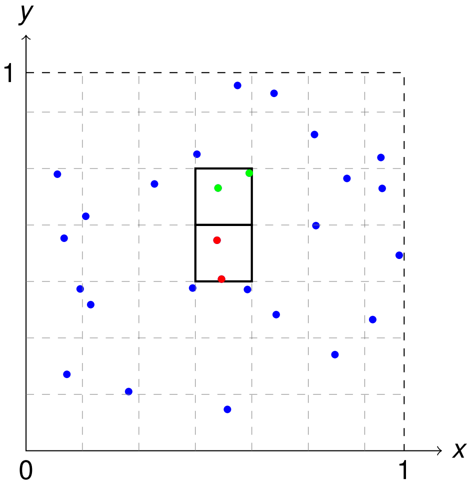
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



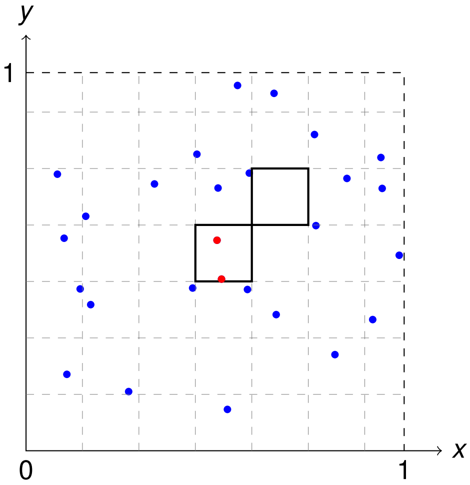
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



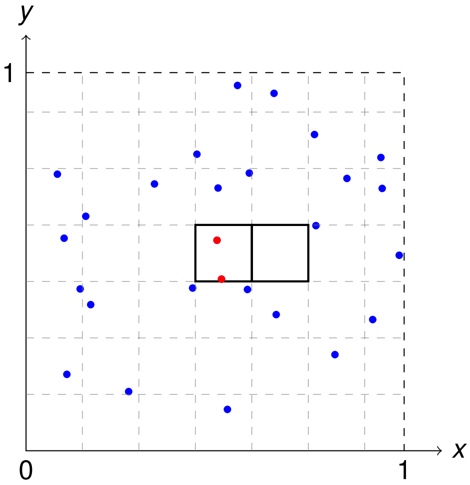
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



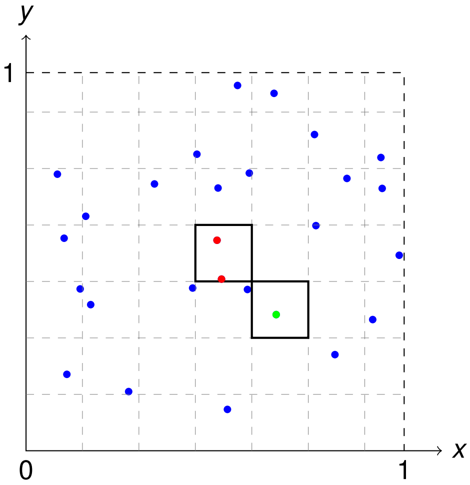
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute

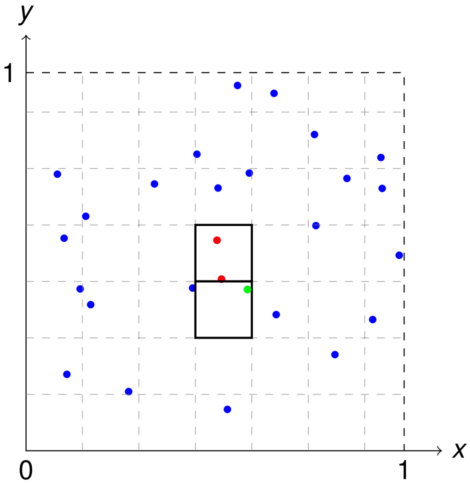


Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



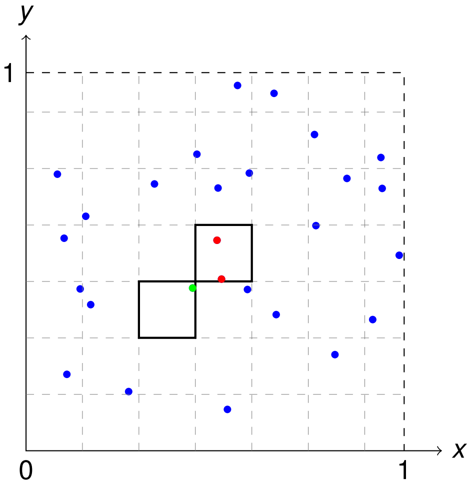
Algorithm



- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute

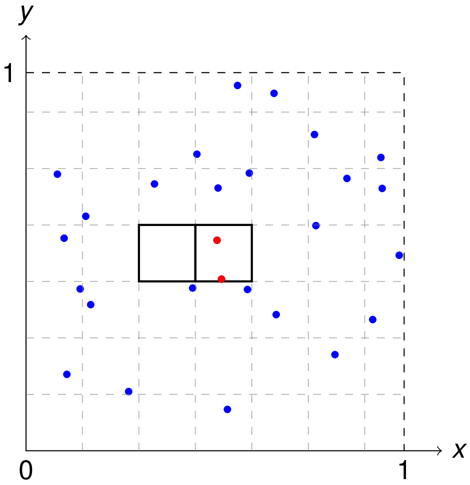
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



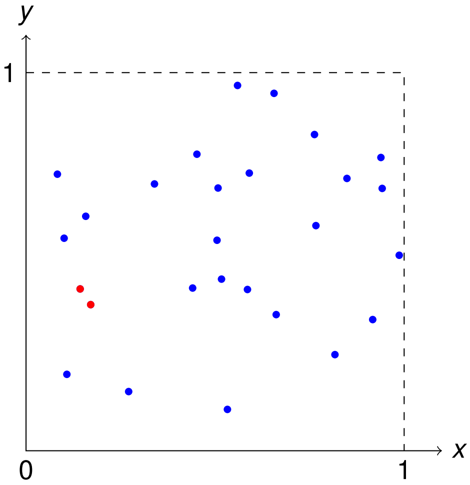
Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



Algorithm

- Sample ($n^{2/3}$ Points from P)
- Partition
- Compute



Sample

- Computer all pair of distances: $O((n^{frac23})^2) = O(n^{frac43})$
- Divide & Conquer: $O(n^{frac23} \log n^{frac23}) = O(n^{frac23} \log n) \subset O(n)$
  More samples possible:
- Other approach?

Partition

- At most n cells contain point
- Map points to their cells, then hashing
- Divide the coordinates by cell length and truncate them to integer.
- $O(n)$

Compute

Another Approach for Step 1

Parallelization

Proof

Reference I

[1] Jon Louis Bentley and Michael Ian Shamos. “Divide-and-conquer in multidimensional space”. In: *Proceedings of the eighth annual ACM symposium on Theory of computing*. STOC ’76. Hershey, Pennsylvania, USA: ACM, 1976, pp. 220–230. DOI: 10.1145/800113.803652. URL: http://doi.acm.org/10.1145/800113.803652.

[2] Martin Dietzfelbinger et al. “A reliable randomized algorithm for the closest-pair problem”. In: *Journal of Algorithms* 25.1 (1997), pp. 19–51.

[3] Steven Fortune and John E Hopcroft. *A note on Rabin’s nearest-neighbor algorithm*. Tech. rep. Cornell University, 1978.

Reference II

[4] Samir Khuller and Yossi Matias. “A simple randomized sieve algorithm for the closest-pair problem”. In: *Information and Computation* 118.1 (1995), pp. 34–37.

[5] J. Kleinberg and E. Tardos. “Algorithm Design”. In: Pearson Education, 2006. Chap. 13 Randomized Algorithms.

[6] Richard J. Lipton. *Rabin Flips a Coin*. 2009. URL: `http://rjlipton.wordpress.com/2009/03/01/rabin-flips-a-coin/` (visited on 07/15/2013).

[7] Michael Oser Rabin. “Probabilistic algorithms”. In: *Algorithms and Complexity: New Directions and Recent Results*. Ed. by Joseph Frederick Traub. Academic Press, 1976, pp. 21–39.

Load Balancing in Step 3

testc

Implementation of Hashing