

Computational Geometry

(csci3250)

Laura Toma

Bowdoin College

Warm-up: Finding collinear points

Problem:

Given a set of n points in 2D, determine if there exist three that are collinear.

Finding collinear points

Brute force

Algorithm 1 (brute force)

- for all distinct triplets of points p_i, p_j, p_k : if collinear return true
- (if you get here) return false

- Questions:
 - Correct?
 - Worst-case running time?
 - Space?

Finding collinear points

Via sorting

Algorithm 2

- initialize array L = empty
- for all distinct pairs of points p_i, p_j
 - compute their line equation (slope, intercept) and add it to an array L
- sort array L by (slope, intercept)
- traverse L and if you find any 3 consecutive identical (s,i) → collinear

- Questions:
 - Correct?
 - Worst-case running time?
 - Space?

Finding collinear points

With a binary search tree

Algorithm 3

- initialize BBST = empty
 - for all distinct pairs of points p_i, p_j
 - compute their line equation (s, i)
 - insert (s,i) in BBST; if when inserting you find that (s,i) is already in the tree, you got three collinear points and return true
 - (if you ever get here) return false
-
- Questions:
 - Correct?
 - Worst-case running time?
 - Space?

Finding collinear points

With hashing

Algorithm 4

- initialize HashTable = empty
- for all distinct pairs of points p_i, p_j
 - compute their line equation (s, i)
 - insert (s,i) in HashTable; if when inserting you find that (s,i) is already in the HT, you got three collinear points and return true
- (if you ever get here) return false

- Questions:
 - Correct?
 - Worst-case running time?
 - Space?

Finding collinear points

A different way to sort

Algorithm 5

- for every point p_i
 - set array $L = \text{empty}$
 - for every point p_j (with $p_j \neq p_i$)
 - * compute slope of p_j wrt to p_i and add it to array L
 - sort L
 - traverse L and if you find two consecutive points that have same slope, they are collinear with p_i so return true
- (if you get here) return false

- Questions:
 - Correct?
 - Worst-case running time?
 - Space?