

Warm-up Question

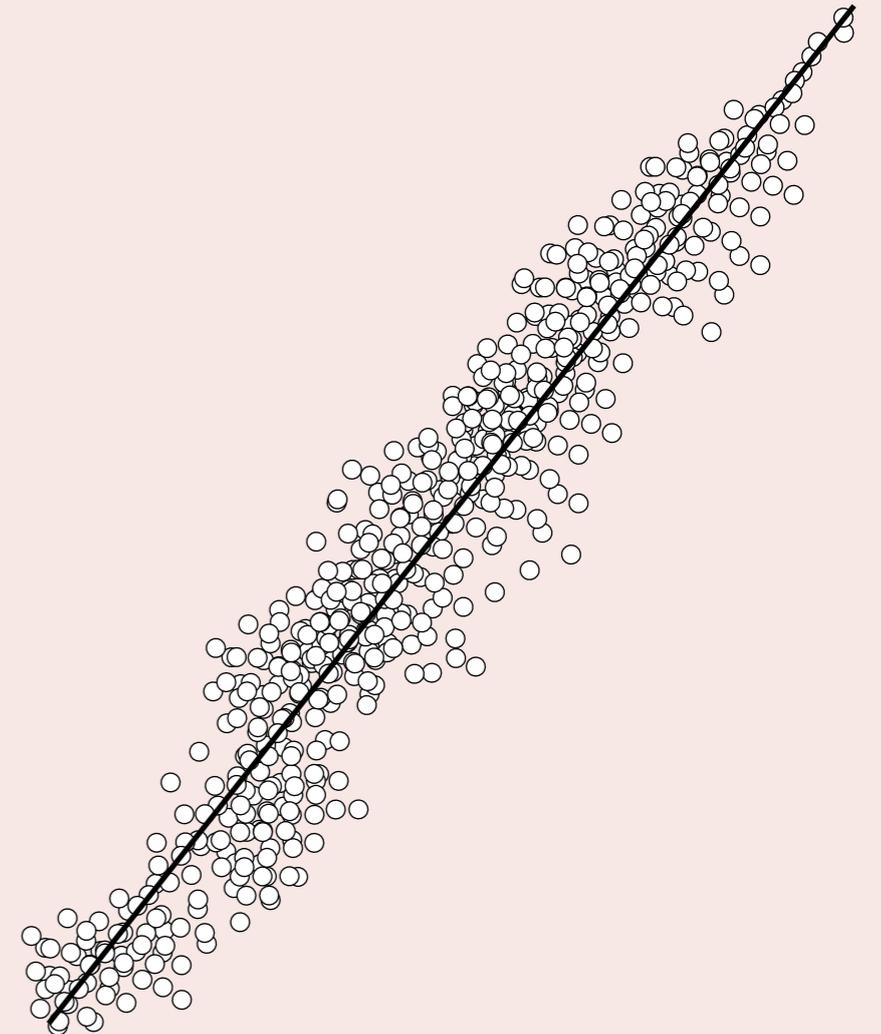
Problem. Given n points in the 2D plane, is there a line that passes through at least $n/8$ of these points?

Naive Solution.

For every pair of points p_1 and p_2 :
COUNT the number of points on the line passing through p_1 and p_2



Running
Time =
 $\Theta(n^3)$



Warm-up Question

Problem. Given n points in the 2D plane, is there a line that passes through at least $n/8$ of these points?

Naive Solution.

For every pair of points p_1 and p_2 :
COUNT the number of points on the line passing through p_1 and p_2


Running Time = $\Theta(n^3)$

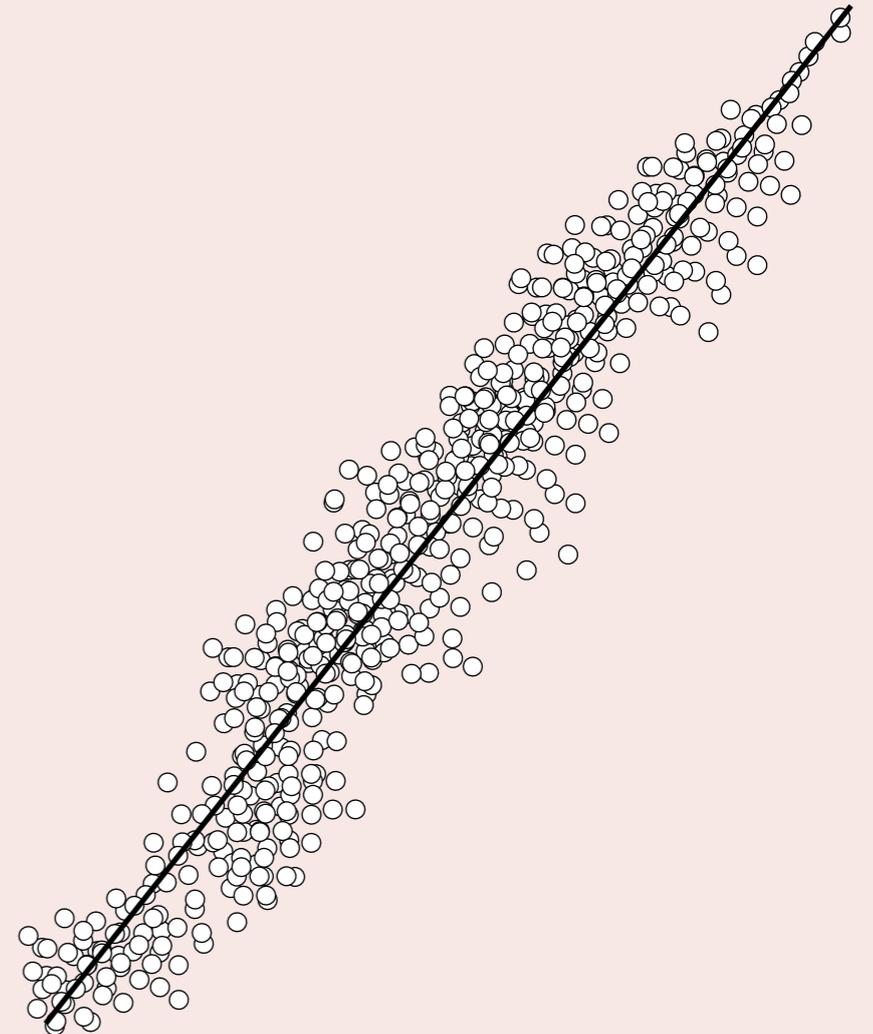
A Fast (but poor) Solution.

CHOOSE 2 random points p_1 and p_2 .

COUNT = number of points on the line passing through p_1 and p_2

IF COUNT $\geq n/8$ **RETURN** TRUE
ELSE **RETURN** FALSE


Running Time = $\Theta(n)$



What is the probability of finding the line by chance?

Let M be a set of collinear points, where $|M| = n/8$

$$P(p_1 \in M) = 1/8$$

$$P(p_2 \in M) = 1/8$$

$$P(p_1 \in M \text{ and } p_2 \in M) = 1/64$$

I.e. Probability of not finding the line = $63/64 \approx 98.44\%$ 

Boosting Up The Poor Solution

Problem. Given n points in the 2D plane, is there a line that passes through at least $n/8$ of these points?

A Fast (but poor) Solution.

```
CHOOSE 2 random points p1 and p2.
```

```
COUNT = number of points on the  
line passing through p1 and p2
```

```
IF COUNT  $\geq$   $n/8$  RETURN TRUE
```

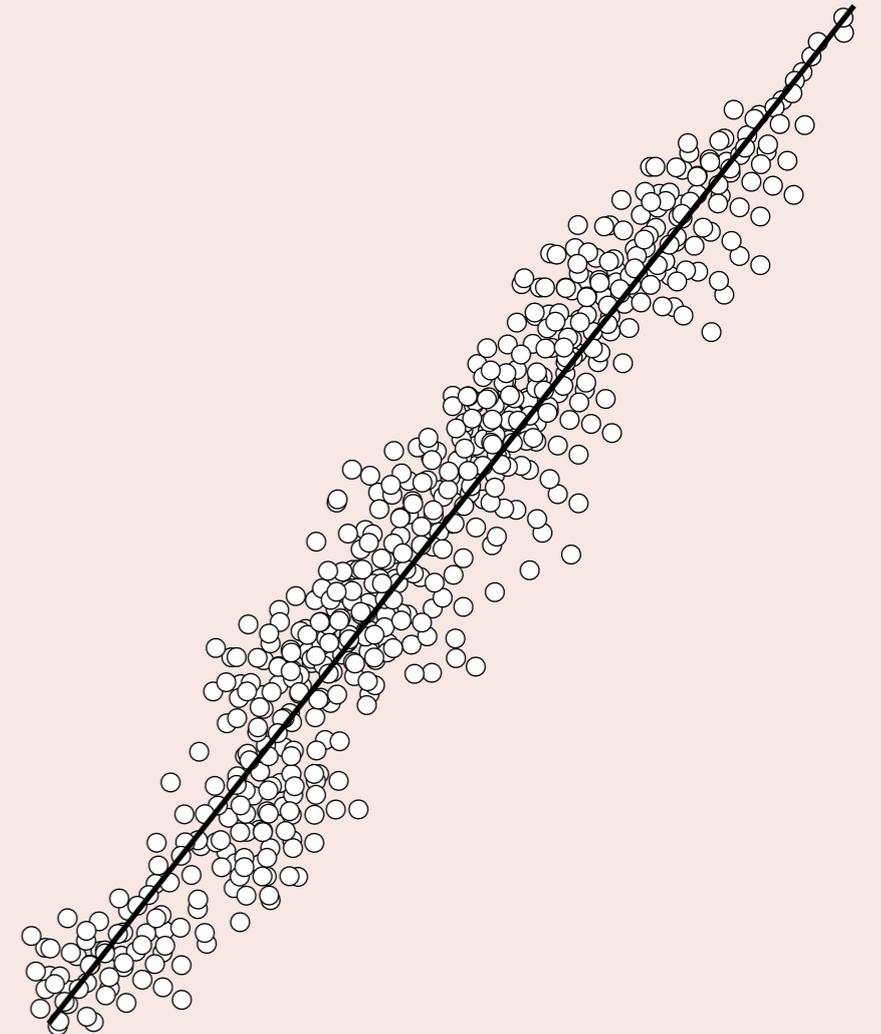
```
ELSE RETURN FALSE
```

A Little Bit of Cheating. Repeat the above solution **400** times. Return FALSE only if no iteration returns true.

What is the probability that all 400 iterations will not find the line?

$(0.9844)^{400} \approx 0.18\%$

I.e. The algorithm will almost certainly find the line!



Boosting Up The Poor Solution

Problem. Given n points in the 2D plane, is there a line that passes through at least $n/8$ of these points?

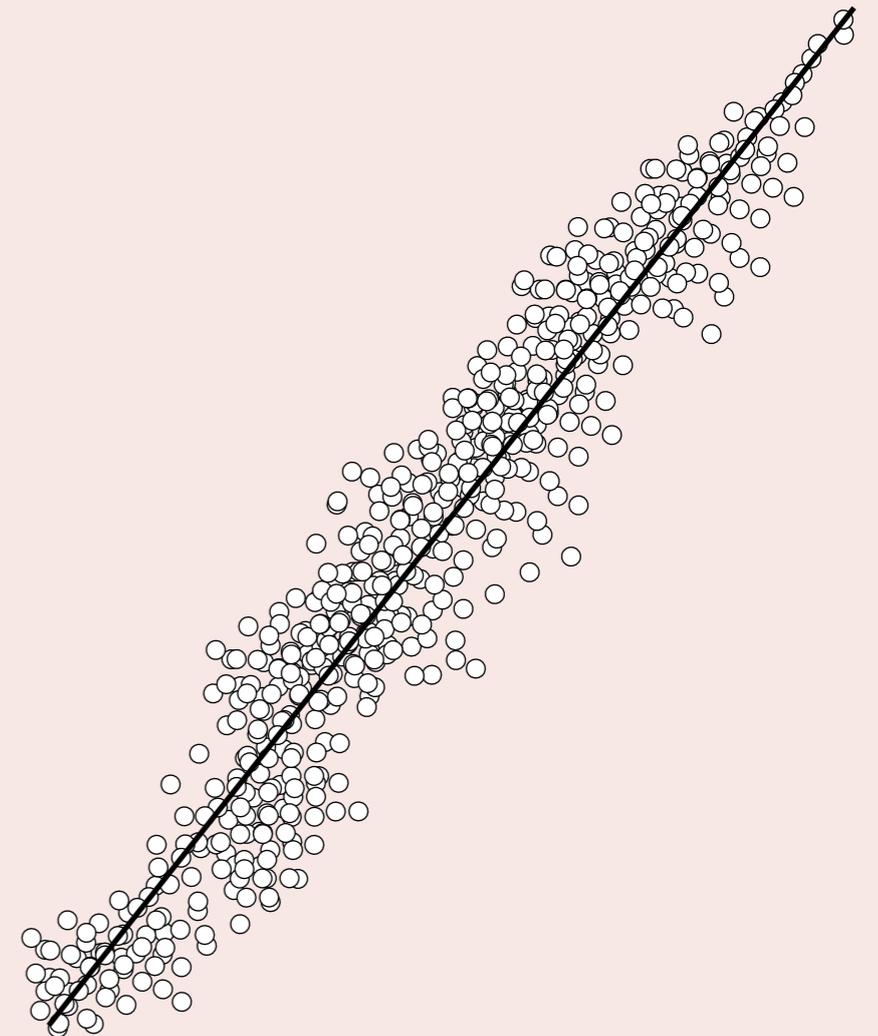
A Fast (but poor) Solution.

```
CHOOSE 2 random points p1 and p2.
```

```
COUNT = number of points on the  
line passing through p1 and p2
```

```
IF COUNT  $\geq$   $n/8$  RETURN TRUE
```

```
ELSE RETURN FALSE
```



A Little Bit of Cheating. Repeat the above solution **400** times. Return FALSE only if no iteration returns true.

What is the probability that all 400 iterations will not find the line?

$$(0.9844)^{400} \approx 0.18\%$$

I.e. The algorithm will almost certainly find the line!

A Useful Trick.

If the probability that A is incorrect = 99% then repeating A 450 times makes the probability that all the runs are incorrect = $0.99^{450} \approx 1\%$

Types of Randomized Algorithms

Monte Carlo Algorithms

- Have a deterministic running time.
- Not guaranteed to be correct (because of the use of randomness).
- Useful Monte Carlo algorithms are efficient and either find the correct solution with high probability or provide a very good approximation of it.

Examples

- Algorithm in the previous slide.
- Stay tuned for more!

Las Vegas Algorithms

- Always give the correct results.
- Their running time can vary (because of the use of randomization).
- Useful Las Vegas algorithms have good *expected* running times.

Examples

- Quicksort and quickselect with shuffling or random sampling of pivots.
- Stay tuned for more!