Introduction
CSCI 532
Closest Pair Problem

Given $n$ points, find a pair of points with the smallest distance between them.
Closest Pair Problem

Solution 1:

1. Compute distance for each pair.
2. Select smallest.
Closest Pair Problem

Solution 2:

1. Split in half.
2. Find closest in left and right sides.
3. Find closest straddling middle.
4. Select closest of all three.
Solution 3:

1. Consider points in random order.
2. Let $\delta =$ closest pair found so far.
3. For each new point, check all “close” points for one $< \delta$.
4. If found update $\delta$. 

\[
\{ p_1, p_2, p_3, \ldots, p_n \} \\
\delta = d(p_1, p_2)
\]
Closest Pair Problem

Solution 3:
1. Consider points in random order.
2. Let $\delta = \text{closest pair found so far}$.
3. For each new point, check all “close” points for one $< \delta$.
4. If found update $\delta$.

Solution 1:
1. Compute distance for each pair.
2. Select smallest.

Solution 2:
1. Split in half.
2. Find closest in left and right sides.
3. Find closet straddling middle.
4. Select closest of all three.
Closest Pair Problem

Solution 3:
1. Consider points in random order.
2. Let $\delta$ = closest pair found so far.
3. For each new point, check all “close” points for one < $\delta$.
4. If found update $\delta$.

Solution 2:
1. Split in half.
2. Find closest in left and right sides.
3. Find closest straddling middle.
4. Select closest of all three.

Solution 1:
1. Compute distance for each pair.
2. Select smallest.

What algorithm is best?
Closest Pair Problem

Solution 3:
1. Consider points in random order.
2. Let $\delta$ = closest pair found so far.
3. For each new point, check all "close" points for one $< \delta$.
4. If found update $\delta$.

Solution 2:
1. Split in half.
2. Find closest in left and right sides.
3. Find closest straddling middle.
4. Select closest of all three.

Solution 1:
1. Compute distance for each pair.
2. Select smallest.

What algorithm is best?
It depends.
Closest Pair Problem

Solution 3:
1. Consider points in random order.
2. Let $\delta$ = closest pair found so far.
3. For each new point, check all “close” points for one $< \delta$.
4. If found update $\delta$.

Solution 2:
1. Split in half.
2. Find closest in left and right sides.
3. Find closest straddling middle.
4. Select closest of all three.

Solution 1:
1. Compute distance for each pair.
2. Select smallest.

What algorithm is best?
It depends.

What does it mean for one algorithm to be “better” than another?
Closest Pair Problem

Solution 3:
1. Consider points in random order.
2. Let $\delta$ = closest pair found so far.
3. For each new point, check all “close” points for one $< \delta$.
4. If found update $\delta$.

Solution 2:
1. Split in half.
2. Find closest in left and right sides.
3. Find closest straddling middle.
4. Select closest of all three.

Solution 1:
1. Compute distance for each pair.
2. Select smallest.

What algorithm is best?
It depends.

What does it mean for one algorithm to be “better” than another?
Possible metrics: Running time, accuracy, resource requirements, simplicity, non-randomized.
Closest Pair Problem

Solution 3:
1. Consider points in random order.
2. Let $\delta$ = closest pair found so far.
3. For each new point, check all "close" points for one $< \delta$.
4. If found update $\delta$.

Solution 2:
1. Split in half.
2. Find closest in left and right sides.
3. Find closest straddling middle.
4. Select closest of all three.

Solution 1:
1. Compute distance for each pair.
2. Select smallest.

What algorithm is best?
It depends.

What does it mean for one algorithm to be “better” than another?
Possible metrics: Running time, accuracy, resource requirements, simplicity, non-randomized.

532 Goals:
**Closest Pair Problem**

**Solution 3:**
1. Consider points in random order.
2. Let $\delta$ = closest pair found so far.
3. For each new point, check all “close” points for one $< \delta$.
4. If found update $\delta$.

**Solution 2:**
1. Split in half.
2. Find closest in left and right sides.
3. Find closest straddling middle.
4. Select closest of all three.

**Solution 1:**
1. Compute distance for each pair.
2. Select smallest.

---

**What algorithm is best?**
It depends.

**What does it mean for one algorithm to be “better” than another?**
Possible metrics: Running time, accuracy, resource requirements, simplicity, non-randomized.

**532 Goals:**
Tools, tools, tools.
Closest Pair Problem

Solution 3:
1. Consider points in random order.
2. Let $\delta$ = closest pair found so far.
3. For each new point, check all "close" points for one $< \delta$.
4. If found update $\delta$.

Solution 2:
1. Split in half.
2. Find closest in left and right sides.
3. Find closest straddling middle.
4. Select closest of all three.

Solution 1:
1. Compute distance for each pair.
2. Select smallest.

What algorithm is best?
It depends.

What does it mean for one algorithm to be “better” than another?
Possible metrics: Running time, accuracy, resource requirements, simplicity, non-randomized.

532 Goals:
Tools, tools, tools.
Tools to build algorithms. Tools to analyze algorithms. Tools to compare algorithms. Tools to share algorithms.

4. Select closest of all three.
Skills Check

• Not graded.
• Allows me to see where we are starting.