What is Comp. Geometry?
- algs for geom probls
- lots of appln's
- SoCG

Possible Topics
- geom. data structures

Point location

in 1D:

O(n) space by sorting
O(log n) query time

in 2D? more challenging
dynamic?

range search

in 1D, ( )

O(n) space
O(log n) time to count
In 2D?

- Nearest neighbor search

  - Dynamic? $O(\log^2 n)$ update
  - $O(\log n)$ query time

Approximate?

Geometric approx alg's:
- Geom. set cover
- Geom. independent set
- Geom. TSP

Combinatorial geometry
- Lower envelope of line segments
Orthogonal Range Searching

Store $n$ pts in $\mathbb{R}^d$ in data structure $S$. Given query range $q$, find pts in $q$. axis-aligned rectangle/box

diff versions:
- report all,
- decide emptiness
- count
- sum of weights
- max weight
  - (median weight, most freq. weight, report all colors)
Space $O(n)$
preproc time $O(n \log n)$ by sorting
Query time $O(\log n)$ for counting
$O(\log n + k)$ for reporting
(output size)
(optimal in comparison model)

2D?

Method 1: "k-d Tree"

Space $O(n)$
preproc. $P(n) = 2P(\frac{n}{2}) + O(n)$
median-finding

$\Rightarrow O(n \log n)$

query algm, given rectangle $q$:

if $q$ does not intersect node’s cell) return 0
else if $q$ completely contains node’s cell
return # pts in cell
It a overlaps
else if a completely contains node's cell
return # pts in cell
else recurse on both children
return sum

```
\[
\begin{array}{c}
\text{easy} \\
\text{tougher}
\end{array}
\]
```

**Analysis:**
query time = \(O(\#\text{cells visited})\)
= \(O(\#\text{cells crossing } 2q)\)

boundary has \(4\) line segments

**How many cells in k-d tree can a vertical line cross?**

```
\[
\begin{array}{c}
1 \text{ of } 2 \\
\text{good}
\end{array}
\quad
\begin{array}{c}
2 \text{ of } 2 \\
\text{bad}
\end{array}
\]
```

\[f(n) = 2f(n/2) + 1\]
\[\Rightarrow O(n) \text{ bad!}\]

```
\[
\begin{array}{c}
1 \text{ of } 4 \\
\text{good}
\end{array}
\quad
\begin{array}{c}
2 \text{ of } 4 \\
\text{bad}
\end{array}
\]
```

\[f(n) = 2f(n/4) + 1\]
\[\Rightarrow O(n^{\log_4 2})\]

\[\Rightarrow O(n^{1.69})\]
query time $O(4 \cdot f(n))$

$= O(\sqrt{n})$

($+k$ for reporting)

Higher $d$:

$f(n) = 2^{\frac{d-1}{2}} f(\frac{n^{1/2d}}{2^d}) + 1$

$\Rightarrow O(n^{\frac{1}{d}}) = O(n^{1 - \frac{1}{3d}})$