CS260 – Lecture 12

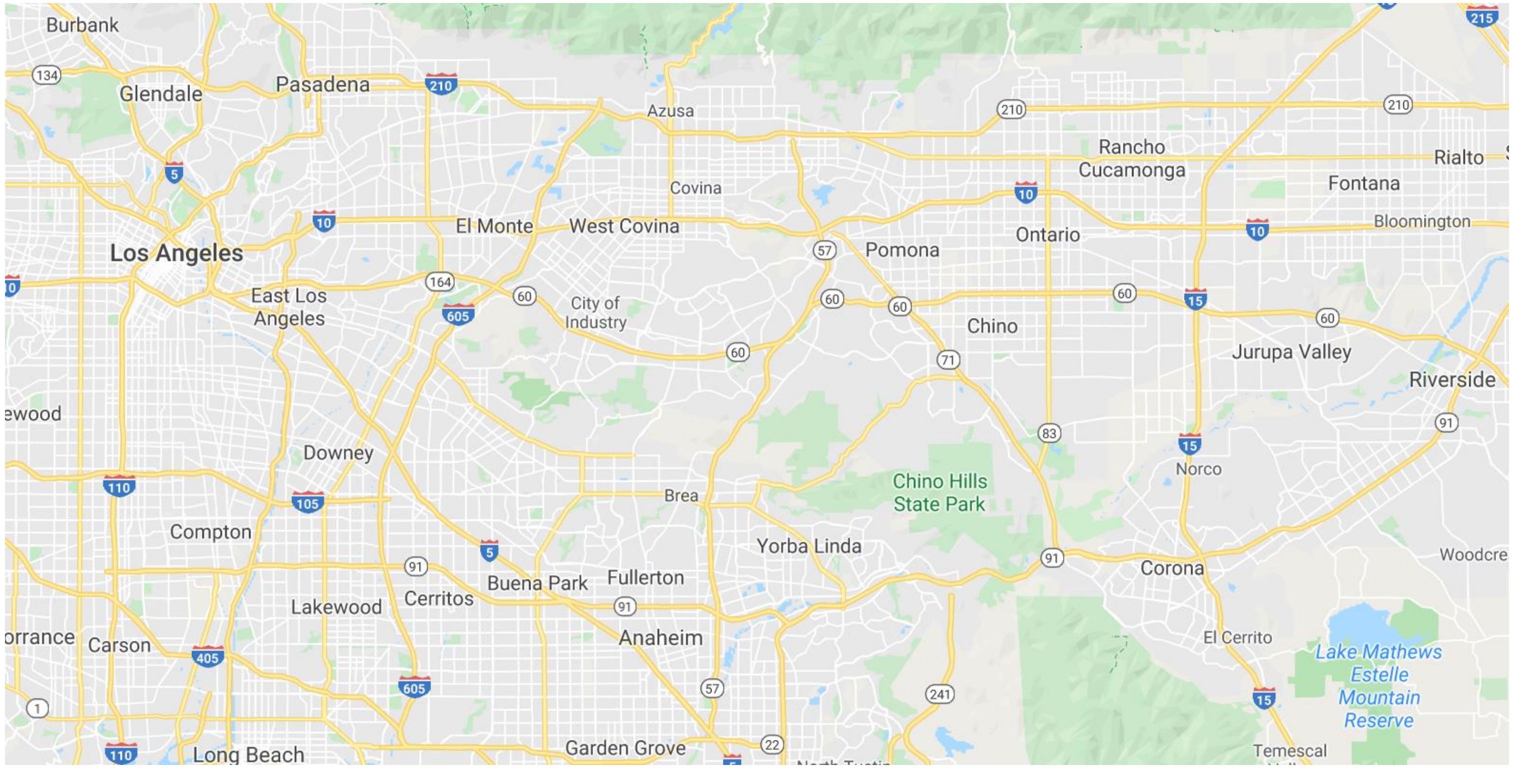
Yan Gu

Algorithm Engineering
(aka. How to Write Fast Code)

An Overview of Geometry Processing

What is geometry processing?

- Graph studies the relationship of objects
- Geometry studies the locations of the objects themselves

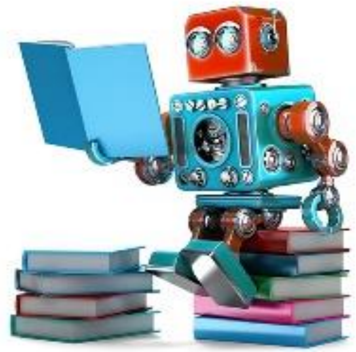


Lots of real-world applications

Every area requires geometry processing

Database / Data warehouses

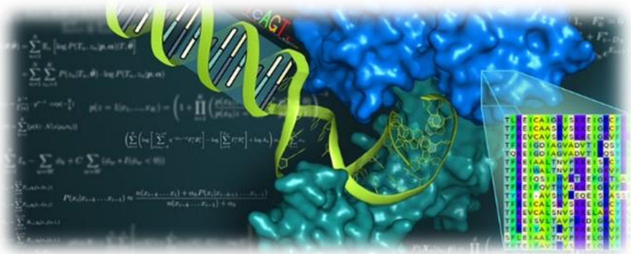
Data mining / Data science



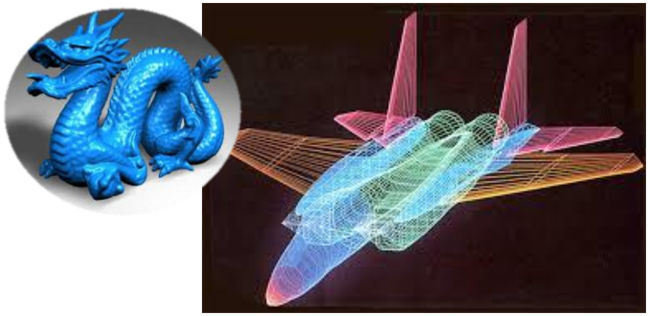
Machine learning / Artificial intelligence

Geometric Information Systems (GIS)

Computational biology



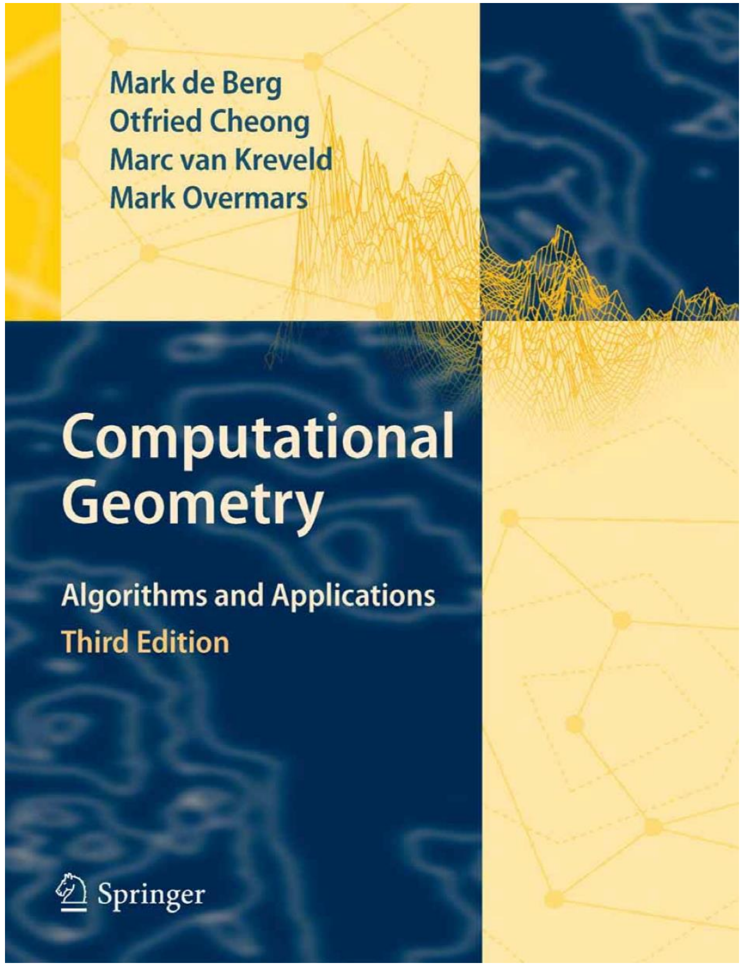
Computer graphics



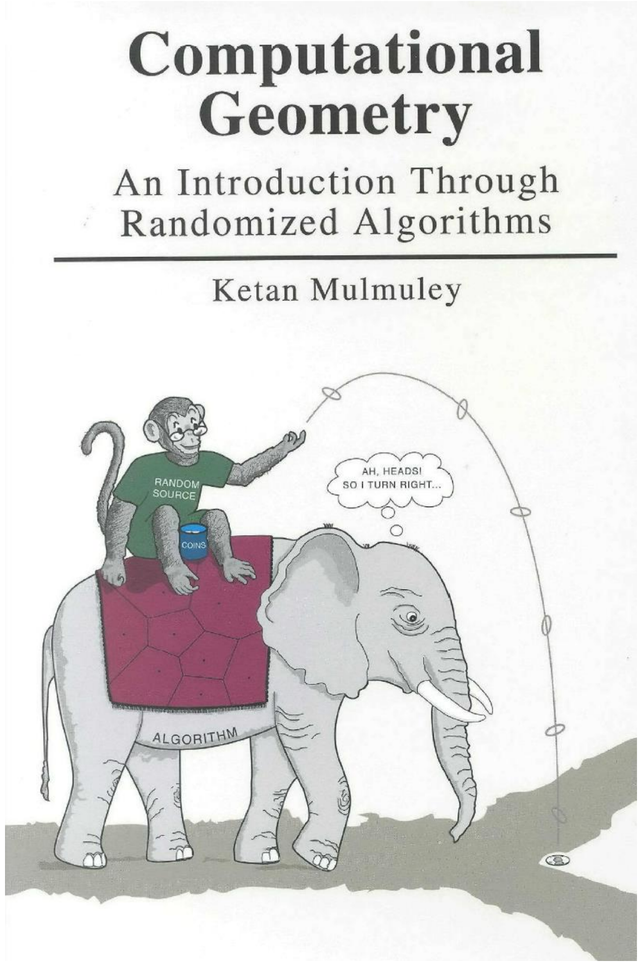
Computational Geometry

- Started in mid 70's
- Focused on abstracting the geometric problems, and design and analysis of algorithms for these problems
- Many problems well-solved, while many other problems remain open
- UCR CS 133 – Computational Geometry
- MIT 6.838 – Geometric Computation

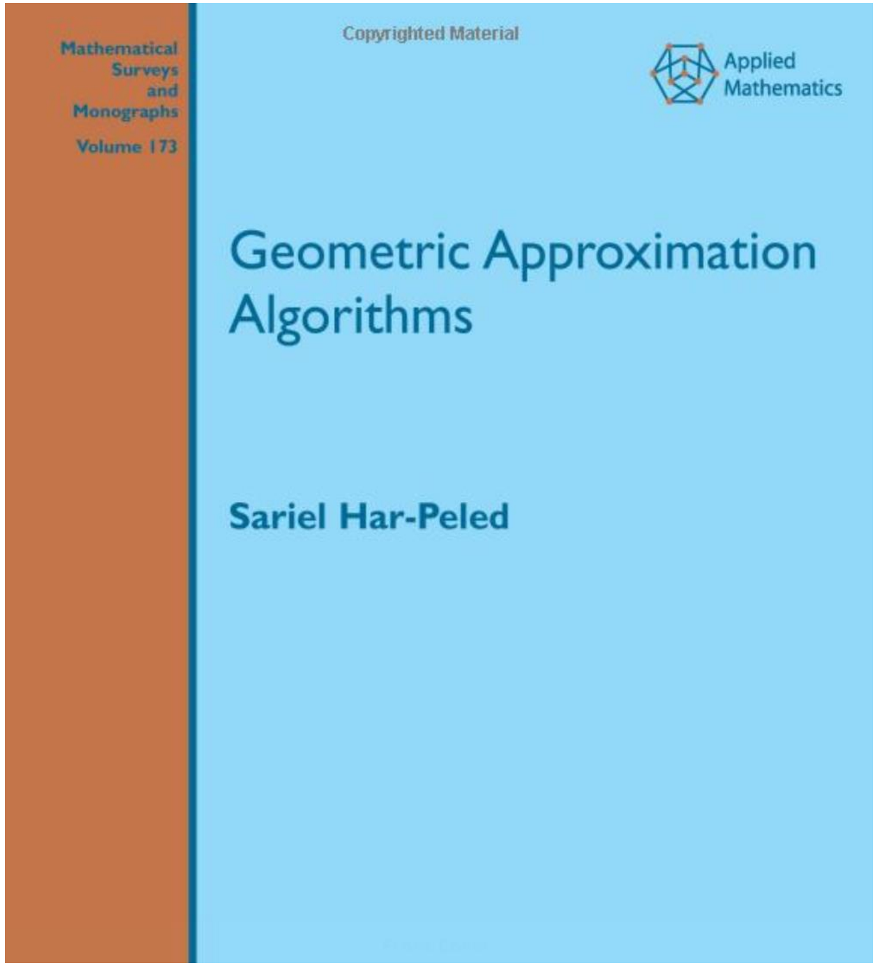
Good references for sequential geometric algorithms



“Four Dutchmen”

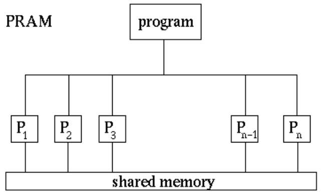


“The elephant book”

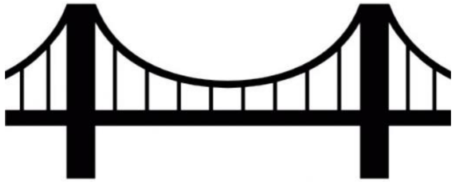


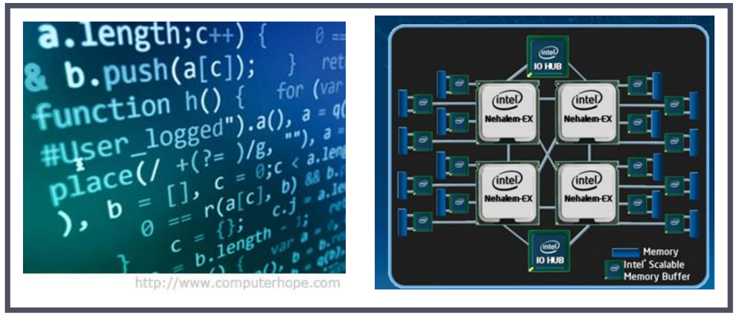
Har-Peled’s book

Algorithm Engineering



O(n log n)
O(n)
O(log n)





Theory

Practice

- Theory is no longer sufficient to guarantee good theoretical performance, because computer architecture becomes significantly more sophisticated

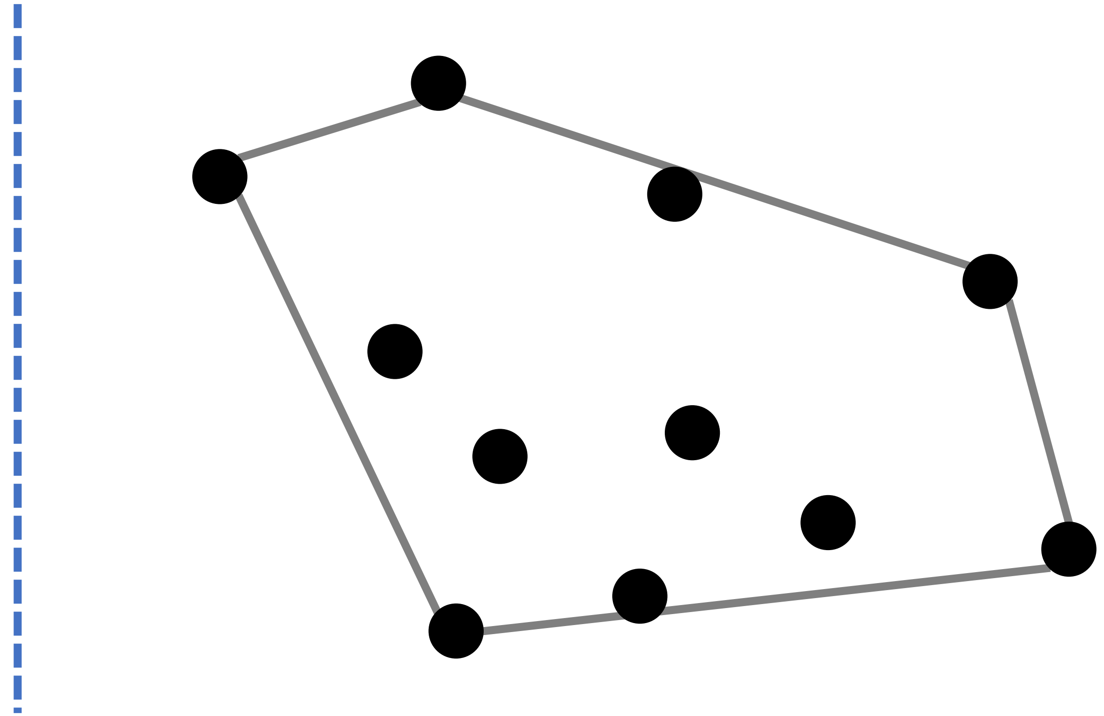
- Parallelism, I/O efficiency, new hardware such as non-volatile memories, and specific applications

Convex Hull, Triangulation, and others

Convex hull

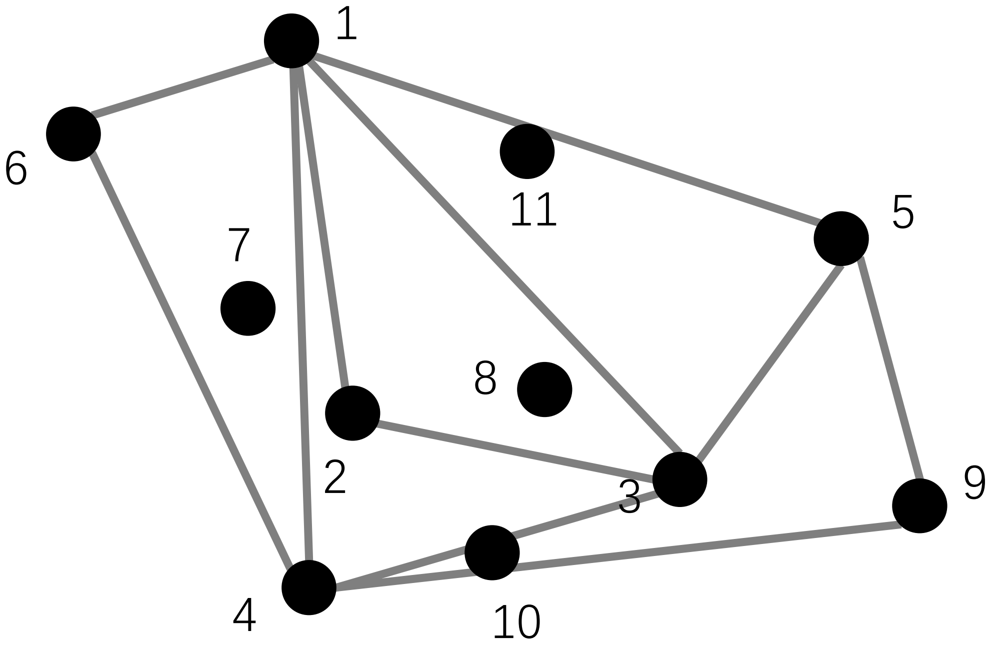
- **A set is convex if every line segment connecting two points in the set is fully contained in the set**

- **Example applications: nearest/farthest point to a line, point**

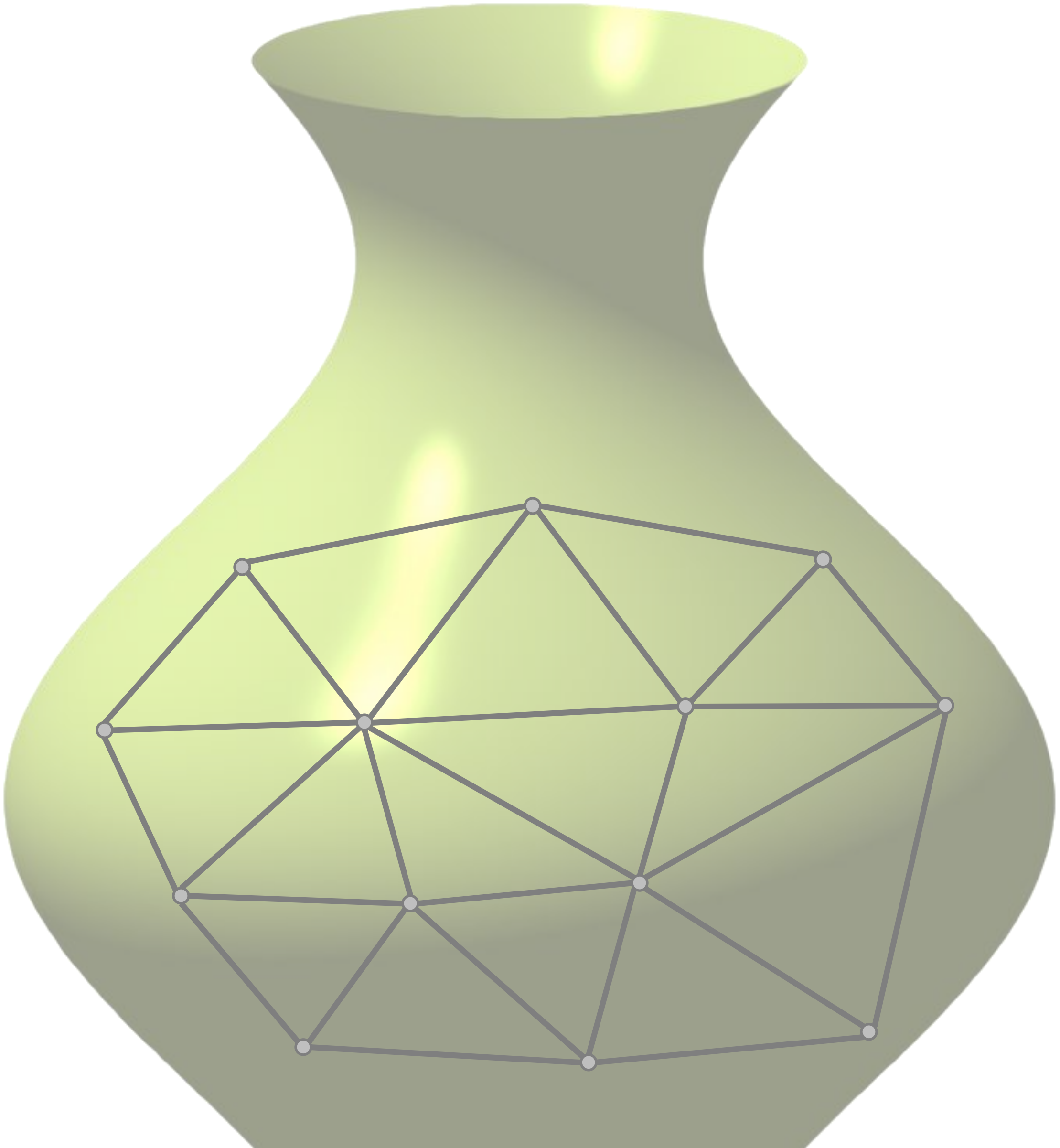


How to construct a convex hull

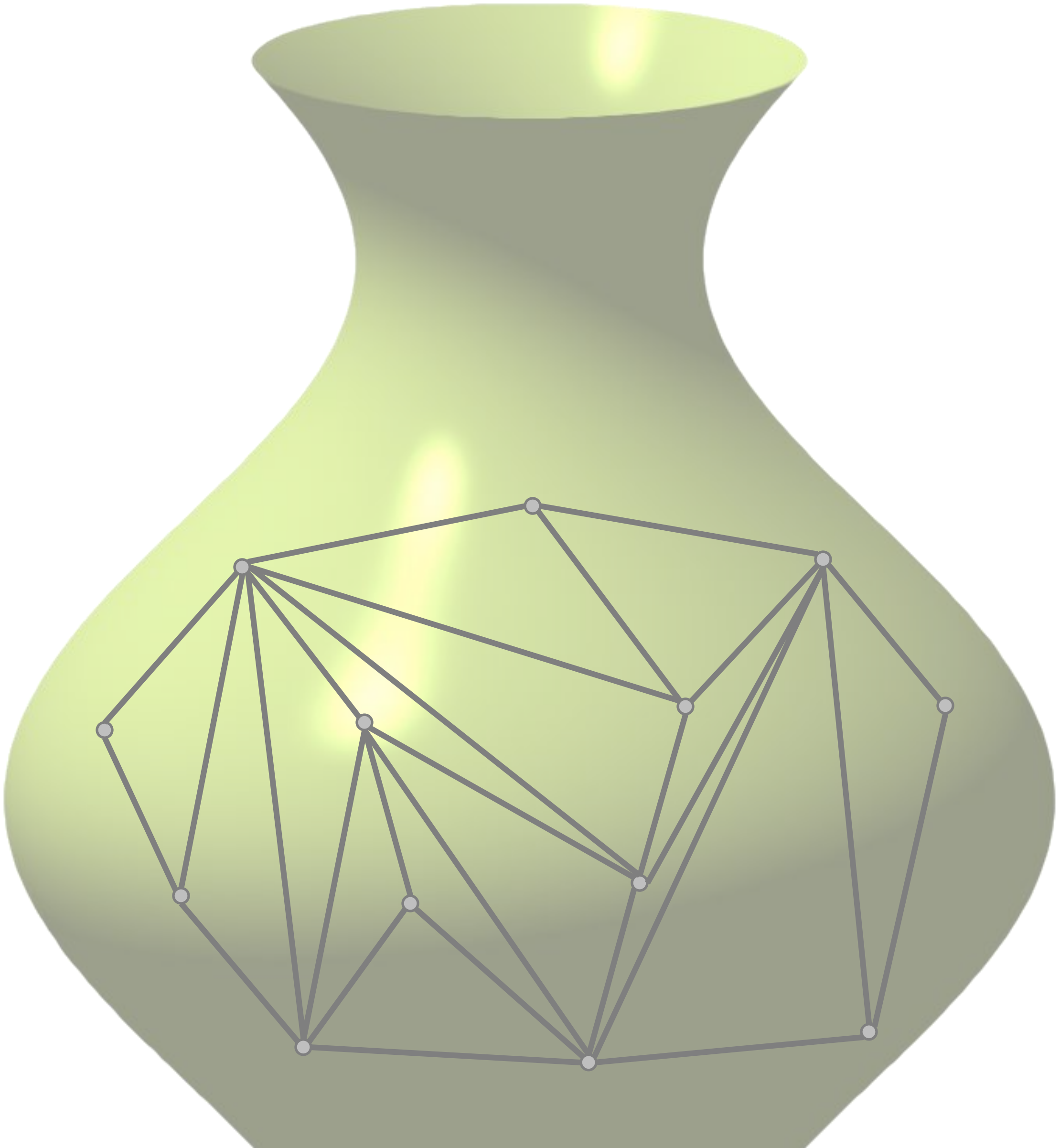
- **Dozens of algorithms sequentially, a few parallel ones for 2D**
- **Randomized incremental construction**
- **Only requires $O(\log n)$ rounds if adding all possible points**
  [BGSS SPAA 2020]



What's triangulation for?

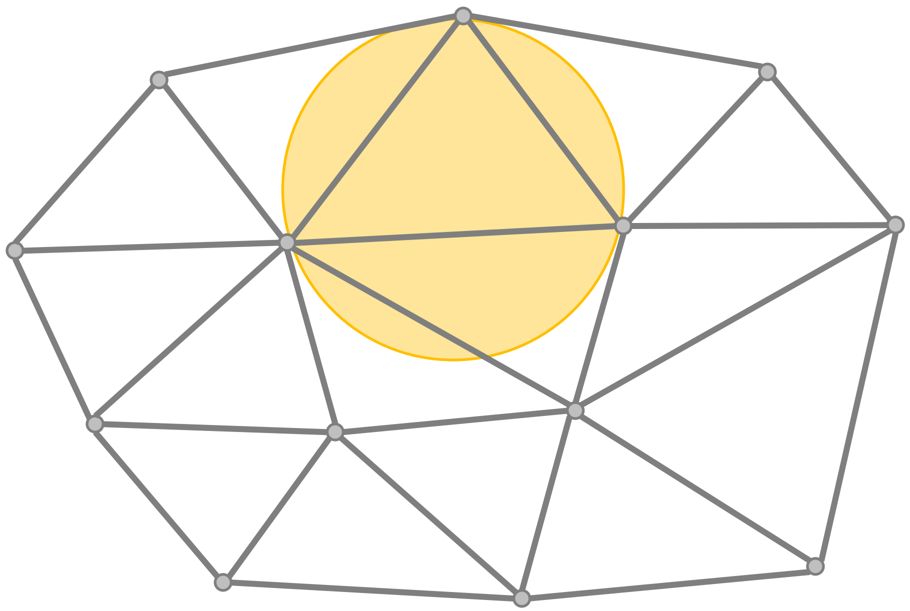


What's a good triangulation?



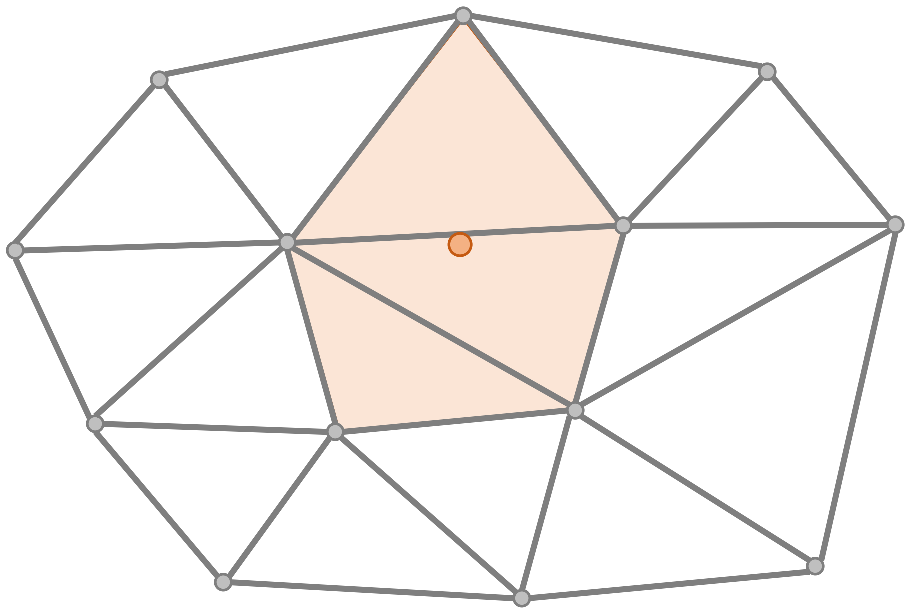
Delaunay triangulation (DT)

- **No points are in the circumcircle of each triangle**
- **Getting practical parallel algorithms for DT is notoriously hard**



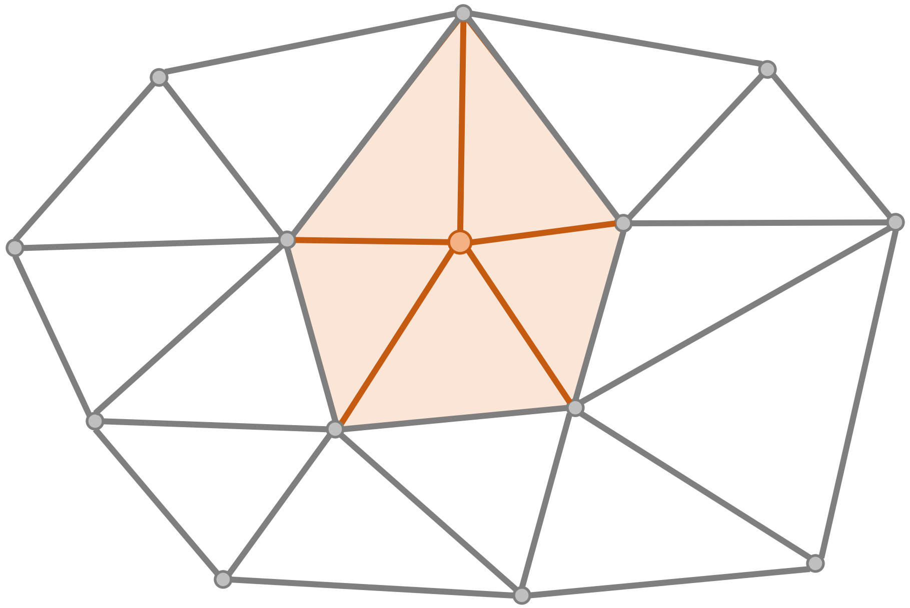
Delaunay triangulation (DT)

- **Again, consider the incremental construction**



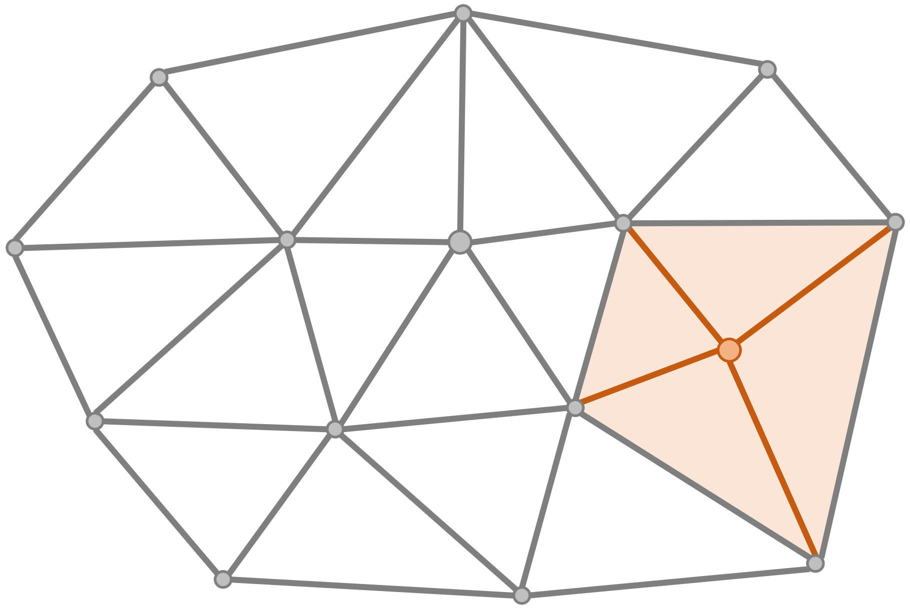
Delaunay triangulation (DT)

- **Again, consider the incremental construction**



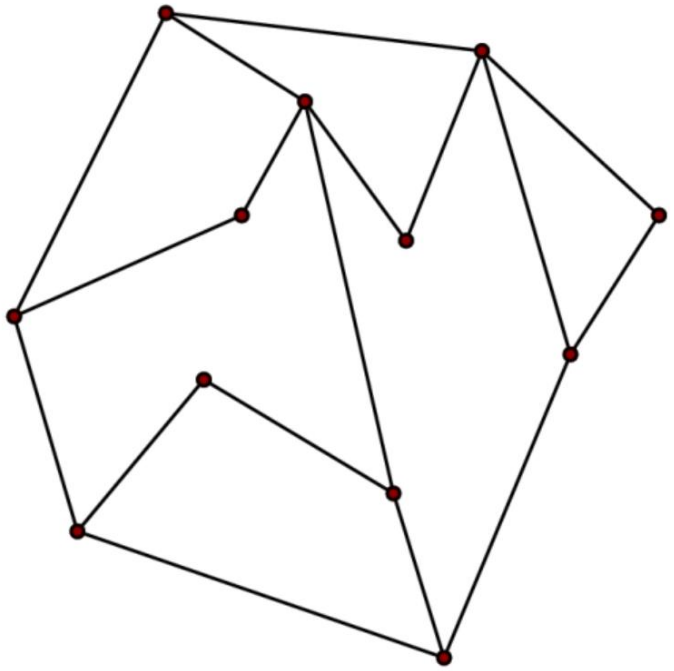
Delaunay triangulation (DT)

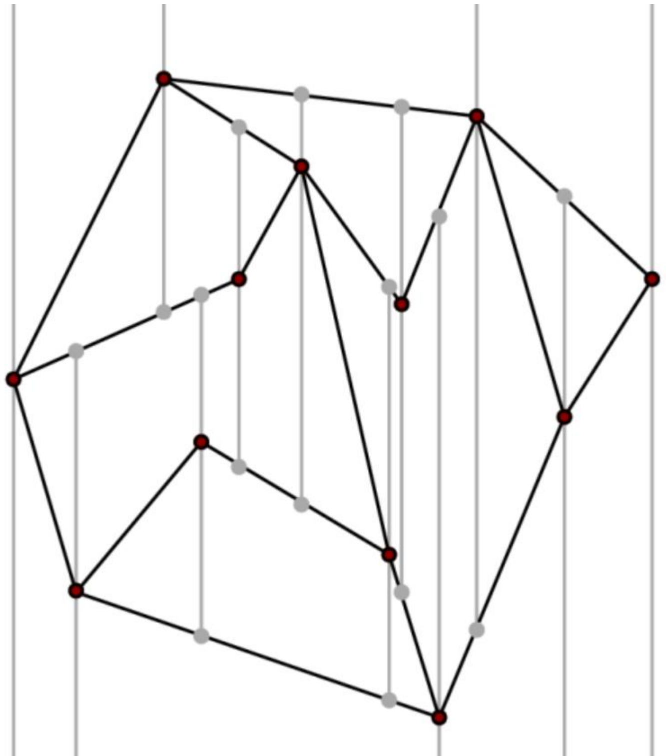
- **Again, consider the incremental construction**
- **2D parallel version is given in** [BGSS SPAA 2016 / JACM 2020]
- **Higher-D version is in** [BGSS SPAA 2020], **since k-D DT can be subsumed by $(k+1)$-D convex hull**



Point location

- **In a 2D plain, decide which polygon a query point belongs to**
- **Example solution: trapezoidal decomposition (no parallel solution)**
- **Other options: Delaunay refinement**





Range Searches

Example range searches

- Nearest neighbor search
- k-nearest neighbor (kNN) search
- Near neighbor counting
- Rectangular range search
- Rectangle queries
- Three-sided queries
- Segment (stabbing) queries

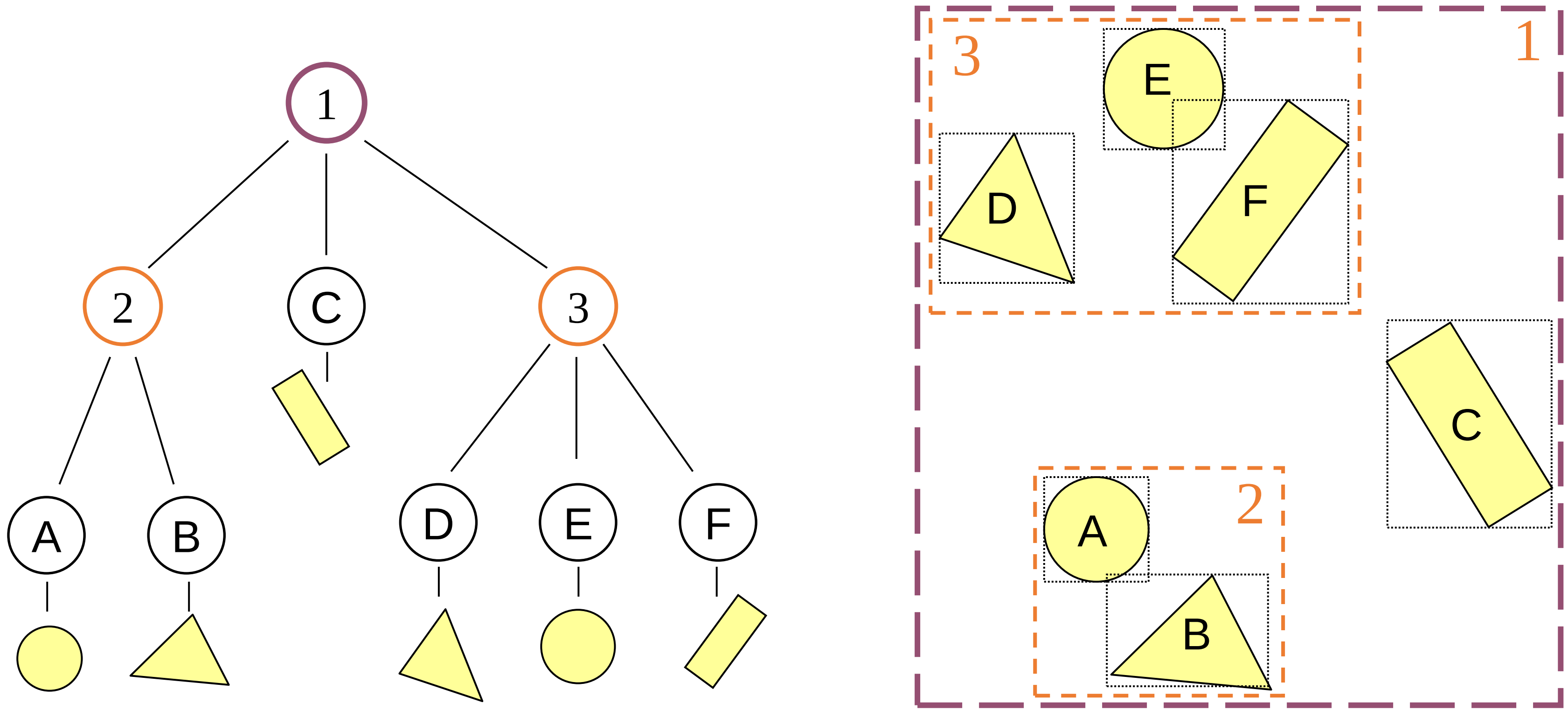
- Ray-scene intersection queries
- Collision detection
- etc.

Building blocks in other algorithms / systems

Classic (sequential) data structures

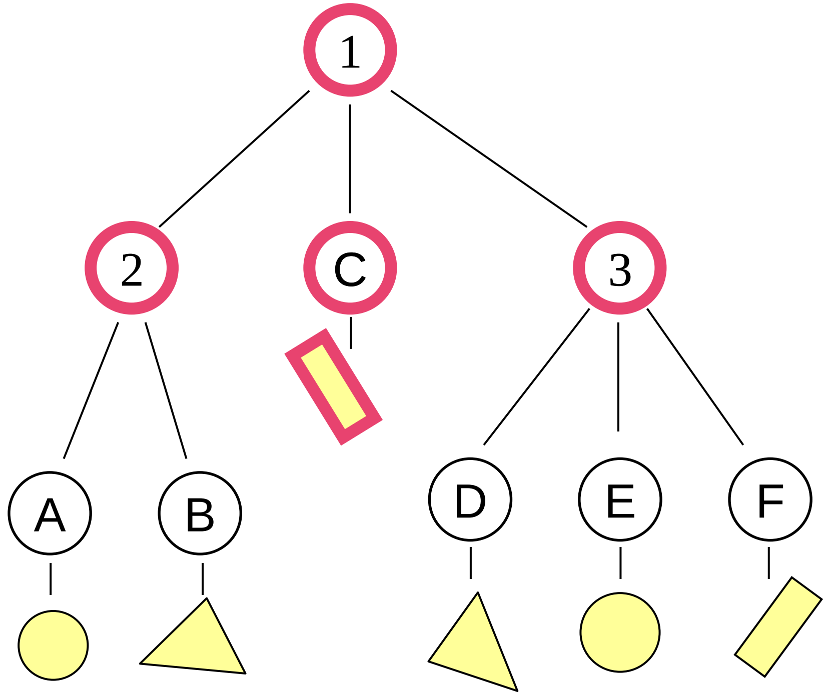
- **Quad/octree**
- ***k*-d tree**
- **Interval tree**
- **Segment tree**
- **Range tree**
- **Priority tree**
- **Other augmented trees**
  - R-tree, bounding volume hierarchy (BVH)

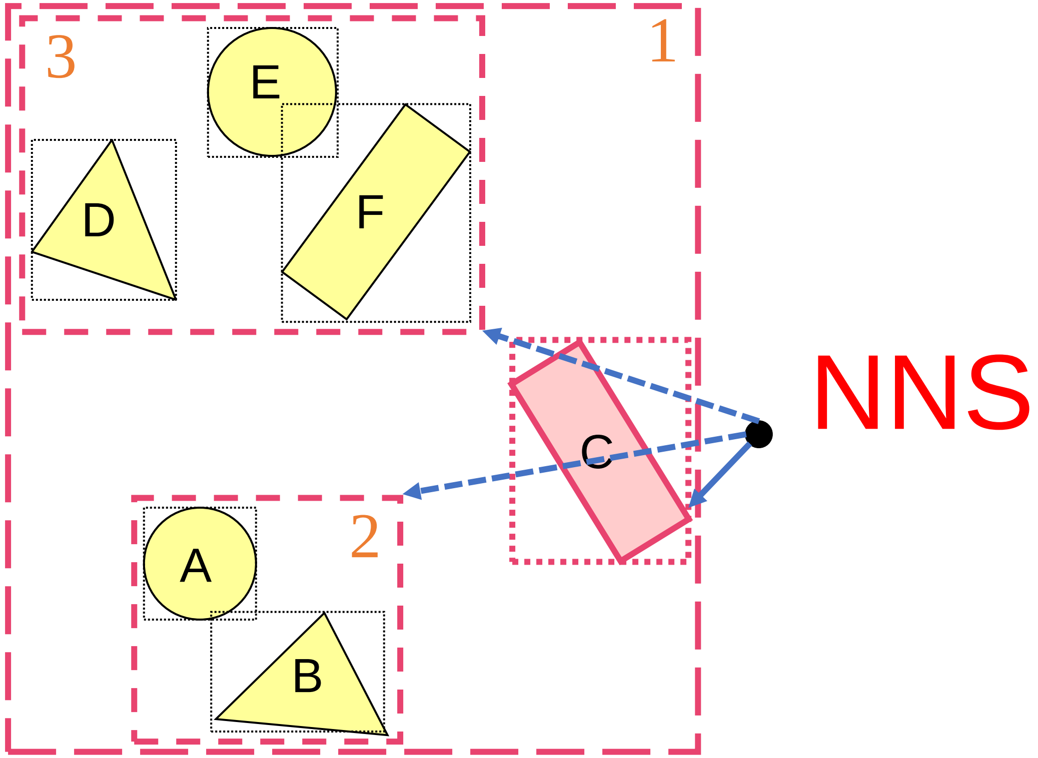
Why trees?



Each interior tree node acts as a fast-pass check for the subtree nodes

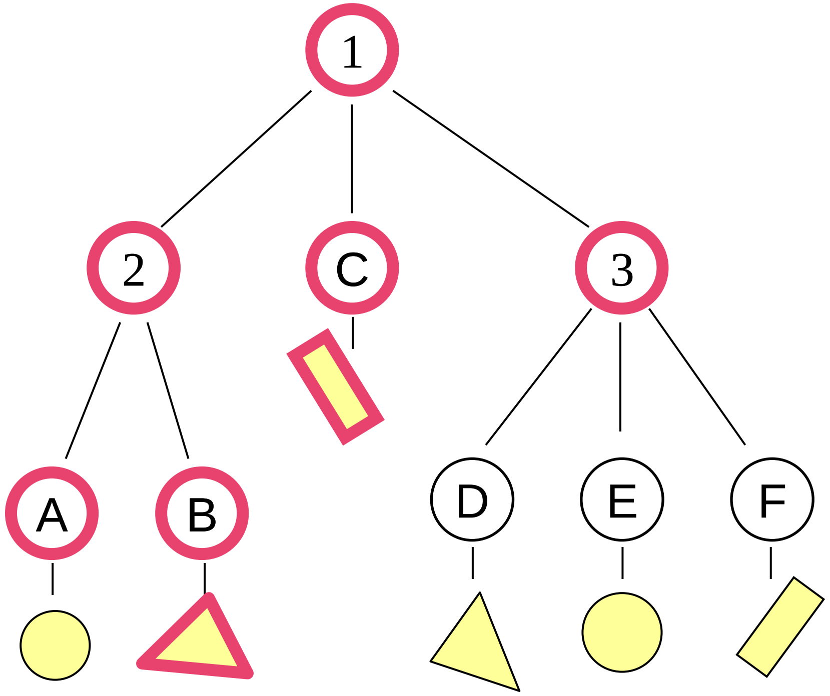
- **No need to traverse the subtree if the traversal can be pruned**

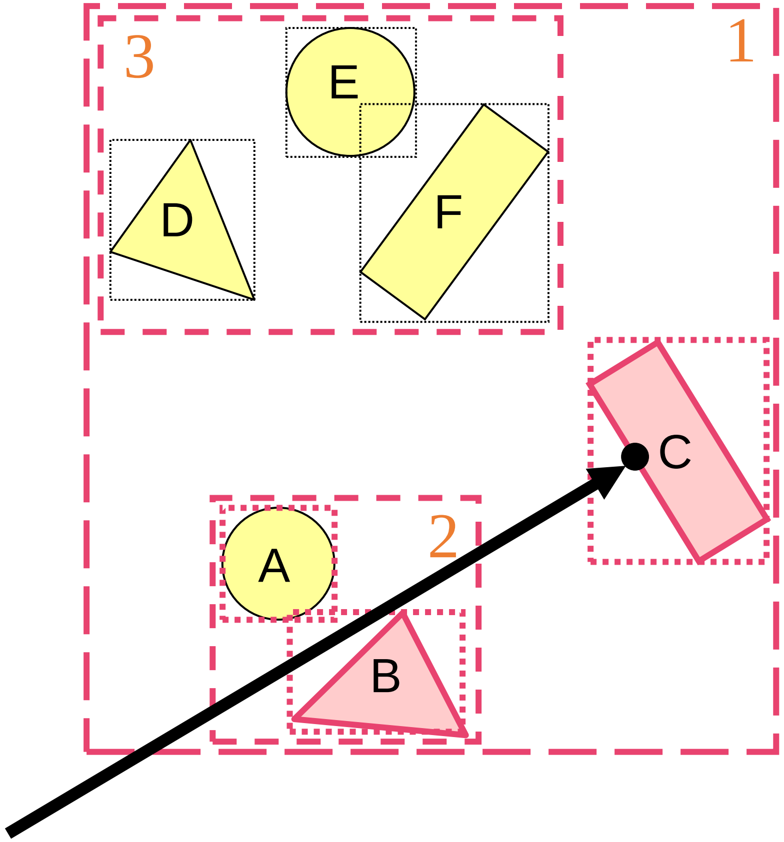




Each interior tree node acts as a fast-pass check for the subtree nodes

- **No need to traverse the subtree if the query misses the bounding box**





Example range searches

- Nearest neighbor search
- k-nearest neighbor (kNN) search
- Near neighbor counting
- Rectangular range search
- Rectangle queries
- Three-sided queries
- Segment (stabbing) queries
- Ray-scene intersection queries
- Collision detection
- etc.

Classic (sequential) data structures

- **Quad/octree**
- **k-d tree**
- **Interval tree**
- **Segment tree**
- **Range tree**
- **Priority tree**
- **Other augmented trees**
  - R-tree, bounding volume hierarchy (BVH)

“Modern” Geometry Problems

Problems that are less “classic” (from MIT 6.838)

- Clustering
- Range searches in “high” (>5) dimensions
- Low-distortion embeddings

- Geometric algorithms for modern architecture
- Geometric algorithms for streaming data

Monday lecture (5/18)

- **Efficient BVH Construction via Approximate Agglomerative Clustering, by Lei Zhang**
- **Parallel Range, Segment and Rectangle Queries with Augmented Maps, by Longze Su**

Wednesday lecture (5/20)

- **Theoretically-Efficient and Practical Parallel DBSCAN, by Jinyang Liu**
- **Engineering a compact parallel Delaunay algorithm in 3D, by Haide He**

Friday lecture (5/22): final project milestone

- **Every of you will give a 5-minute talk about the current progress of your final project (with slides)**
- **Reserve your time slot here: https://docs.google.com/spreadsheets/d/1De2HOpzUewLK6laMJqnxF5bahzSBu1fJUR_hSMM9qt8/edit?usp=sharing**

Paper reading is due this Friday (5/15)