CPSC 516 - 20.09.22
Class IV

Admin

1. HW 1 out, due 29 September.
2. Think of HW as including "Talk to Will about a project idea".
3. Scribing has begun!

Convex Hulls

Figure 1: Convex Hull Example

A set S of points is *convex* if $\forall p, q \in S$, the points on $\overline{pq}$ are also in S.

The *convex hull* of a set Q is the intersection of all convex sets that contain Q. Equivalently, it's the smallest convex set containing Q.

Note A, B convex $\implies A \cap B$ is convex.

Problem given n points in $\mathbb{R}^2$, find the convex hull of the points.

Various Nice Algorithms

- Divide and Conquer (Split in two, take the hull of each, connect with bridge edges)
- Incremental (grow left to right)
- Both $O(n \lg n)$.

Two Old Algorithms (Which we then combine)

Jarvis March / Giftwrapping (1973):

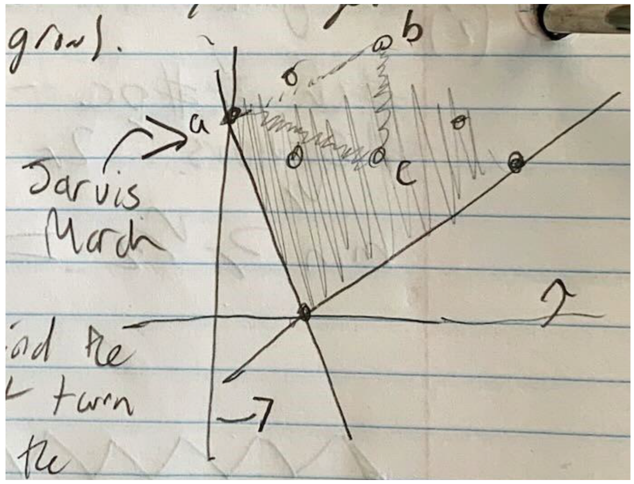

Figure 2: Jarvis March after first two vertices picked

Repeatedly find the next 'min' left turn until you get back to the starting point / the leftmost point. The runtime is $O(nh)$, where h is the number of hull points.

Use the signed area of Δabc to see if there's a left turn from a to b to c.

Graham's Scan (1972):

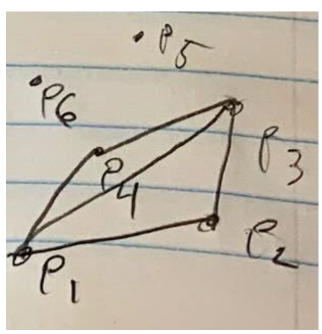

Figure 3: Graham's Scheme after step 4.

1. Find a point $p_1 \in Q$ with smallest y-coordinate.
2. Sort $Q \setminus \{p_1\}$ ccw (counterclockwise) around p_1, and label the remaining vertices $p_2, \ldots, p_n$ according to this sort.
3. Start with $\Delta p_1 p_2 p_3$ as the convex hull of the first 3 points. Put them onto a stack S.

4 … n. While there's no left turn to p_i from the top of the stack, pop S. Then push p_i onto S.

In Search of Optimality

If we only measure the point set by its size, then Graham's Scheme is optimal. However, Jarvis March is better than Graham's Scheme when the hull is small ($h \notin \Omega(n)$). With this in mind, we want to make an algorithm which is optimal with respect to both n and h.

The original optimal algorithm, which runs in $O(n \log h)$ time, is from Kirkpatrick-Seidel. It's quite complicated though, so we'll look at Timothy Chan's algorithm, which is asymptotically identical **(Next class)**.

Lower Bound

Remember the lower bound on sorting is $O(n \lg n)$.

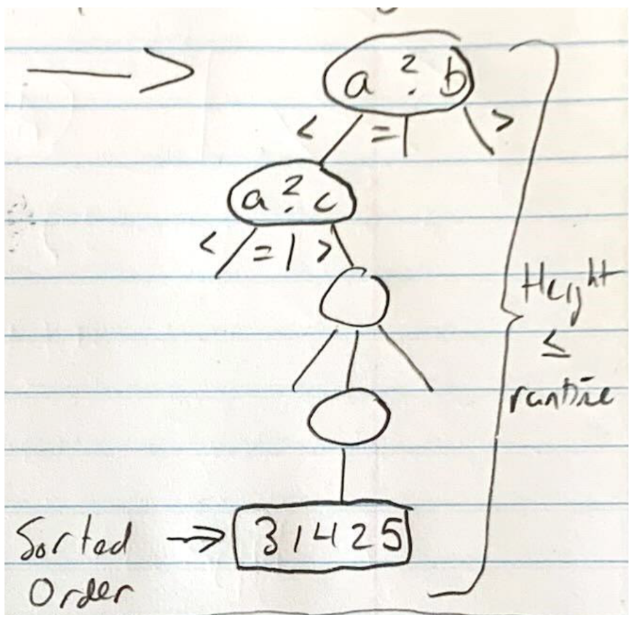

Figure 4: Decision Tree

There are $n!$ possible sorted order, so a decision tree representing the sort has $n!$ leaves. This means its height is $\geq \ln(n!) \in \Theta(n \lg n)$.

We will now reduce sorting to a convex hull problem. Take n numbers, and project them onto the parabola to get the points (i, i^2) (Or equivalently project onto $[0, \pi]$ radians).

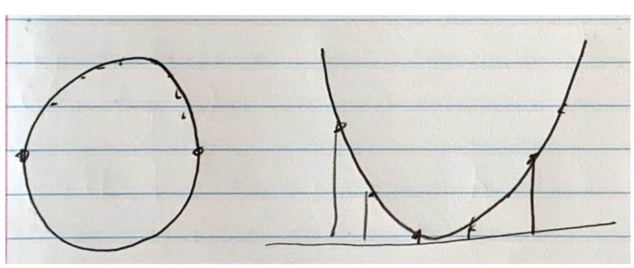

Figure 5: Circle and Parabola projection

Then the resultant convex hull will be all the points in counterclockwise order, so if you start at the smallest point the remainder will appear in sorted order.

However convex hull requires more complex computations than the sorting model allows, so our lower bound for sorting doesn't help us here. We need to introduce a more complicated model.