11 Closest Points, Polynomial Multiplication/FFT

- Divide and conquer
- Closest points
- Fast Fourier transform

11.1 Closest Points

In one dimension, the closest points problem can be solved in $O(n \log n)$ by sorting all points and scanning linearly.

To simplify the problem, we assume that no two points share x or y coordinates.

The main idea here is to recursively subdivide the points into left and right halves (by sorting with respect to the x coordinates), solve the closest pair problem on each half, then check if there is a closer pair with one point on the left and one point on the right subdivision.

As we recurse, we maintain lists sorted by x and y coordinates.

Notation: Input points are $P = \{p_1, \cdots, p_n\}$, where $p_i = (x_i, y_i)$.

Let Q be the points in P with the first $\lceil n/2 \rceil$ x coordinates.
Let R be the points in P with the last $\lfloor n/2 \rfloor$ x coordinates.

We recurisvely find the closest points in Q (call them q_0^*, q_1^*), and R (call them r_0^*, r_1^*)

Let $\delta = \min\{\text{dist}(q_0^*, q_1^*), \text{dist}(r_0^*, r_1^*)\}$.
Let x^* be the largest x coordinate of a point in Q
Let L be the line $x = x^*$.

Lemma 11.1
If $\exists q \in Q$ and $r \in R$ with $\text{dist}(q, r) < \delta$, then both q and r are within δ of L.

With this information, we can restrict our attention to a "strip" centered on L with width 2δ. Call the strip S.

Lemma 11.2
If two points in S have distance less than δ, then they are within 15 positions of each other in the list of points sorted by y coordinates.

Proof. Divide S into squares of side length $\frac{\sigma}{2}$ (arranged in a 4×4 grid).

No two points can be in the same square. If they were, they would have distance at most $\sqrt{2}\frac{\delta}{2} = \frac{\delta}{\sqrt{2}} < \delta$, but δ was the distance between the closest pair of points solely on the left or right, so this can not occur.

Suppose two points in S are 16 or more positions apart in the list sorted by y coordinate.

Then they must differ by at least 3 rows of boxes, so they have distance at least $3 \cdot \frac{\delta}{2} > \delta$. ☐

Running time: $T(n) = 2T(n/2) + O(n) \implies T(n) = O(n \log n)$.

11.2 Polynomial Multiplication and the FFT

Problem: Given polynomials $a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$ and $b(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_{n-1} x^{n-1}$
We wish to compute the product $c(x) = a(x)b(x)$.