Sorting and Searching Algorithms

Winter 2023

Some of our important divide and conquer algorithms involve something involving ordering of elements in a list. Besides being good examples of divide and conquer, these algorithms also are useful tools for solving more complicated problems.

1 Sorting

There are several problems that are made much easier by going through the list of inputs in order, perhaps keeping track of some version of the best example seen so far or perhaps keeping two pointers into the list. This can be used to solve problems like:

- Given a set of points in $\mathbb{R}^2$ find the pareto optimal points (that is the points (x, y) in your set so that there is no other points (x', y') with $x' > x$ and $y' > y$).
- Finding the closest pair of points in some set of real numbers.
- For two sets A and B find the number of pairs of $a \in A$ and $b \in B$ with $|a - b| \leq 1$.
- Given two sets A and B find the closest element of A to each element of B.

2 Order Statistics

Computing order statistics is a bit more specialized, but it does have some uses. Particularly, computing medians of a set is often useful in the divide step of divide and conquer algorithms. The idea of breaking the set based on a random pivot is also often useful for algorithms like quicksort.

3 Searching

The fact that you can search a sorted list in logarithmic rather than linear time is hugely important. In fact, it can be seen as one of the most important reasons that you would want to sort your list to begin with. However, the binary search idea where you make some relatively simple measurement that allows you to cut your search space in half, is much more general and has a lot of applications.