Why don't CS profs ever stop talking about sorting?!

1. Computers spend more time sorting than anything else, historically 25% on mainframes.

2. Sorting is the best studied problem in computer science, with a variety of different algorithms known.

3. Most of the interesting ideas we will encounter in the course can be taught in the context of sorting, such as divide-and-conquer, randomized algorithms, and lower bounds.

You should have seen most of the algorithms - we will concentrate on the analysis.

Applications of Sorting

One reason why sorting is so important is that once a set of items is sorted, many other problems become easy.

Searching

Binary search lets you test whether an item is in a dictionary in $O(\lg n)$ time.

Speeding up searching is perhaps the most important application of sorting.

Closest pair

Given n numbers, find the pair which are closest to each other.

Once the numbers are sorted, the closest pair will be next to each other in sorted order, so an $O(n)$ linear scan completes the job.

Element uniqueness

Given a set of n items, are they all unique or are there any duplicates?

Sort them and do a linear scan to check all adjacent pairs.

This is a special case of closest pair above.

Frequency distribution – Mode

Given a set of n items, which element occurs the largest number of times?

Sort them and do a linear scan to measure the length of all adjacent runs.

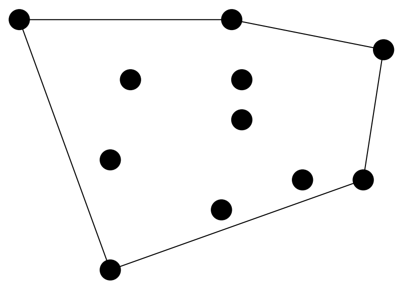
Median and Selection

What is the kth largest item in the set?

Once the keys are placed in sorted order in an array, the kth largest can be found in constant time by simply looking in the kth position of the array.

Convex hulls

Given n points in two dimensions, find the smallest area polygon which contains them all.



The convex hull is like a rubber band stretched over the points.

Convex hulls are the most important building block for more sophisticated geometric algorithms.

Once you have the points sorted by x-coordinate, they can be inserted from left to right into the hull, since the rightmost point is always on the boundary.

Without sorting the points, we would have to check whether the point is inside or outside the current hull.

Adding a new rightmost point might cause others to be deleted.

Huffman codes

If you are trying to minimize the amount of space a text file is taking up, it is silly to assign each letter the same length (ie. one byte) code.

Example: e is more common than q, a is more common than z.

If we were storing English text, we would want a and e to have shorter codes than q and z.

To design the best possible code, the first and most important step is to sort the characters in order of frequency of use.

Character	Frequency	Code
f	5	1100
e	9	1101
c	12	100
b	13	101
d	16	111
a	45	0