Sorted!

Thinking about algorithms. Doing Algoritms

What is sorting?

- Ordering any list or collection in some way
- In-place or out of place?
- Speed is crucial! (Schlemiel...)
- How do we decide what is "bigger"

How??

- Selection sort: Easy to understand but slow... Video
- Bubble sort: Easy, in place and slow. Video
- Quick sort: recursive, elegant, fast
- Merge sort: This is what Javascript's sort() uses

Selection sort

- 4 5 9 7 1 3
- 1 4 5 9 7 3
- 1 3 4 5 9 7
- 1 3 4 5 9 7
- 1 3 5 5 7 9
- 1 3 4 5 7 9
- Sorted.....

Bubble sort

First pass	Second pass	Third pass
4 5 9 7 1 3	4 5 7 1 3 9	4 5 1 3 7 9
4 5 9 7 1 3	4 5 7 1 3 9	4 5 1 3 7 9
4 5 9 7 1 3	4 5 7 1 3 9	4 1 5 3 7 9
4 5 7 9 1 3	4 5 1 7 3 9	4 1 3 5 7 9
4 5 7 1 9 3	4 5 1 3 7 9	4 1 3 5 7 9
4 5 7 1 3 9	4 5 1 3 7 9	4 1 3 5 7 9

Whoa....

- This seems like work
- Selection sort seems faster than bubble, but is it really?
- Which is fastest if the list is already sorted?
- Both are $O(n^2)$ for worst case
- Merge sort is recognised as the fastest ($O(log(n))$ but is quite complex to think about....

Lets play...

- Form teams of 2 people
- Get 6 Post-it notes and write the names of your favourite bands/movies on the Post-it notes
- Now sort them in reverse alphabetic order using a selection sort
- Now do the same using a bubble sort, but alphabetically

Computer task

- Implement an in-place bubble sort on for your list implementation.
- Think about what happens when you have a sorted list (how do you stop)
- Remember that the last element cannot be swapped :)
- There is a new test case on the ListApi Github repo