Week 12.1 — Sorting & Recursion

Recursion — reminder

- Functions call themselves (possibly indirectly)
- Make sure the recursion will stop eventually (base case/s).
- Recursive calls should be to smaller / simpler instances.
- Beware stack overflow.

Why recursion?

“It’s easier”

- . . . to read?
- . . . to write?
- . . . for whom?

Is the person saying it:

- a theoretical computer scientist?
- a fan of “functional programming” languages[1]?
  - Languages without loops = have to use recursion

[1] Lisp, Haskel, Erlang, . . .

Why not iteration?

Can't we use iteration[2] instead?
Sort of.
There is a theoretical result that yes, recursive algorithms can be converted into algorithms which use loops (and a stack).
But, there are some algorithms which really are more clearly expressed in a recursive form.
$F(1) = 1$
$F(2) = 1$
$F(n) = F(n\text{-}1) + F(n\text{-}2)$
Got an iterative Formulation?

[2] loops

Sorting

We'll look at sorting algorithms as an example of a task which recursion can make easier.

Sorting a List in Java

```
List li;
...
li.sort(null);   // use default ordering
```

See Sort.java
Doesn't give us much insight into how it works though.

Merge Sort

Task: Sort `Arr` — an array of N `ints`.

- Suppose we know how to sort $N/2$ ints.
- Split `Arr` in half.
- Sort each half
- Merge the two sorted parts into one sorted whole.

See `MSort.java`, Merge Sort video.

Works because a single element array is automatically sorted.
Sorts “bottom up”.

Quick Sort

Suppose we have a routine Part[3](Arr):

- returns the index of one element in Arr which is guaranteed to be in the correct place.
- everything to the left of that element is $\leq$ than it.
- everything to the right of that element is $\geq$ than it.

See `QSort.java`, Quick Sort video

[3] partition

Which is Better?

It depends:

Criterion	**Better**
Simplicity	Mergesort - easier to understand
Worst case performance	Mergesort - O(n log n)
Expected performance	Quicksort - O(n^2)
Memory requirements	Quicksort - only one array

Why? See an algorithms course.