

More on recursion

Dr Tom Ridge

2018 Q1

- ▶ This is important for sorting, so we do it here
- ▶ Recall that we are happy with while loops, and we are also happy with calling the function again instead of using a while loop
- ▶ Now we will look at recursion proper!
- ▶ The main difference at this stage is that:
 - ▶ with a while loop we can say “now go and do that again”
 - ▶ with recursion we can say “now go and do that again, and when you are finished I want to do something with the result”

Factorial function

Look at the definition (the “recurrence relation” definition) of the factorial function on Wikipedia:

<https://en.wikipedia.org/wiki/Factorial>

Factorial with while

```
// version using while
int fact(int i) {
    int r = 1;
    while(true) {
        if(i==0) return r;
        r = r*i;
        i = i-1;
    }
}
```

Factorial with recursion

The same function, but this is really programmed differently- something different is happening here that you may not have seen before; notice how this exactly matches the recursive defn given in Wikipedia

```
int fact(int i) {  
    if(i==0) return 1;  
    return i*fact(i-1);  
}
```

!!! In class demo of how factorial executes.

A key point: you need to have multiple “execution pointers”, multiple instances of the function.

Fibonacci

https://en.wikipedia.org/wiki/Fibonacci_number

How would you program this in Java, using recursion?

Some more examples

Sigma function link

// <http://www.sparknotes.com/math/algebra2/discretefunction>

```
int sigma(int x) {  
    if(x==0) return 0;  
    return x+(sigma(x-1));  
}
```