Max Gordon

Mr. Shelton

AP Computer Science

21 January 2020

Recursion

Recursion is the process of solving a problem by splitting it into smaller parts. For example if we were to solve a problem like 2 + 2, we can directly evaluate it, there is no need to split it up. On the other hand, lets say we assign f(x) = x + 2. Then we have a problem like f(2) + 2.

To solve it we must:

1. Solve f(2) first
   1. f(2) = 2+2 so f(2) = 4
2. Then solve f(2) + 2
   1. 4 + 2 = 6, because we can substitute f(2) for 4
3. Our final answer is 6

Now let’s make f use itself to calculate its answer, but if we just do f(x) = f(x) + 2, and we try and solve it, it will continue forever because there is no end to it. But what we can do is add a condition, , so when *x* drops below a threshold (6), the recursion stops. To ensure that *x* reaches that threshold, every iteration *x* is continually decreased by a difference of 2. To solve it you must continually subdivide the problem until you get to the base statement, then you can go backwards through all of the subdivisions, and finish.

Let’s say that we solve f(x) for x = 15 using the SSS method (solve, substitute, simplify):

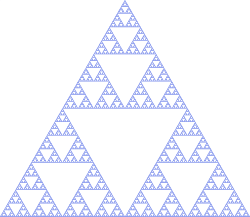
1. - Solve until we get to the base statement

- We are at the base statement, so we must substitute

* + - * 1. - now we will just continue to simplify

So out final answer here will be 64.

Recursion can also be used in geometry; do you see the recursion in this triangle?



It can also be seen in artwork, like this piece from M.C. Escher (I think).

\*for more on recursion, start on page 1