Module 3 - Discussion Prompt Questions
Adam Rich
Section EN.605.202.87
Spring 2018

Describe a scenario where you see recursion at play in your daily life. What are some of the key features of this process or activity that make it recursive; reflect on the defining characteristics in the lecture. For each example, how is this different from an iterative process?

Cleaning the house -> cleaning the downstairs -> cleaning the kitchen -> cleaning the dishes -> cleaning the forks, etc.

Because each step of the process is breaking "the problem" into a set of smaller chunks it is recursive.

Describe a second and different scenario where you see recursion at play in your daily life. What are some of the key features of this process or activity that make it recursive; reflect on the defining characteristics in the lecture? For each example, how is this different from an iterative process?

I know that mowing the lawn was used in the lecture. But, I do it a different way. I mow it is a big circle, starting with the border. If I define the problem as "mow the outer ring" then the recursive process could be: 1. mow the outer ring, if the base case is reached, i.e. no more lawn to mow, mow the outer ring, etc.

Are there any scenarios you can envision in which recursion would really be impossible, that the process is so iterative that you could not look at it recursively at all?

I cannot think of one. I do not believe that there is any problem that cannot be phrased in a recursive way. Like the processes above, it would be easy to phrase those as iterative. One way to look are recursion is to recognize that you are iterating through a set of smaller and smaller problems all of the same type, all with the same base cases.

What are some good, practical ways to organize your thoughts and ideas when working with recursion to manage the process a little more easily?

Know what the base case is or what the base cases are.

Know how you go from a smallish problem to the base case. Don't start with a big problem right away.

Your recursive routine needs to have logic to make the problem smaller. If you are not making the problem smaller before calling the function again, you will never get to the base case.