06/12/2021

Recursion

How function calls work in language.

- ☆ While the f^n is not finished executing, it will remain in stack.
- ☆ When a function finishes executing it is removed from stack, and the flow of program is restored to where f^n was called.

What is Recursion?

f^n calling itself.

Base Condition in Recursion: Condition where our recursion will stop making new calls.

Why Recursion

- ☆ It helps us in solving bigger/complex problems in a simple way.
- ☆ You can convert recursion sol^n into iteration & vice versa.
- ☆ Space complexity is not constant because of recursive calls.
- ☆ It helps in breaking down bigger problems into smaller problems.

How to understand & approach a problem:

1) Identify if you can break down problem into smaller problems
2) Write the recurrence relation if needed.
3) Draw the recursive tree
4) About the tree:
   - ✲ See the flow of f^n, how they are getting in stack.
   - ✲ Identify & focus on left tree calls and Right tree calls.
   - ✲ Draw the tree and pointer again and again using pen & paper
   - ✲ Use a debugger to see the flow.
5) See how the values & what types of values(int, string are returned at each step. See where the function call will come out of. In the end, you will come out of main f^n.

Variables:

① Arguments: will go in next f^n call.
② Body of f^n: Specific to that call.
③ Return: Return what sub recursion f^n are calling.