

Questions

1) Would you find seeing the status of variables from the recursive stack calls helpful?

If so in What visual form would you like to see a stack visualization? I.e. animated, 2D, 3D etc. Why 2D or 3D. What would you like to see in stack visualization?

Yes, both 2D and 3D can work. Must be animated. Like to include information of when stack piles and pops calls on the stack. Maybe 2D - easier to visualize.

Yes, 3D - more fun and looks better. Want a colourful and nice interface.

Yes, 2D is fine - simpler to see things.

Yes, 2D - simple. Basic process of stack good enough.

Yes, anything - 3D. Basic process of stack will be fine.

Yes, 2D animated. Integer being passed to recursive method calls.

Yes, no preference. Just depict the flow of the stack.

Yes definitely. Often times the debugger on Netbeans can be difficult to understand. I would say maybe like a stack of books or some kind of tower that grows with the stack? I think 3D is better because maybe you can represent more information as opposed to 2D. If it was animated it would look better and will probably be more visually appealing. I'd like to see the status of the variables passed to the function each time.

I think students who have a difficult time tracing what their code is doing will find it helpful. But it would have to be very simplified because students who struggle with understanding this may also struggle with interpreting a call stack. We were taught to think about it like a stack of plates so maybe something like that? 3D definitely would look better. I'd like to see the parameters passed to each call stack at the very least.

It depends on the question. I think for harder or more complicated questions it can be helpful for bug fixing. Although I can see it being useful for learning how recursion works internally. A stack of something like blocks or similar would be a good representation of it. 2D or 3D I wouldn't really have a preference. Maybe an option for both? If it was animated nicely with smooth animations and an option to view the stack progress slowly.

Yes I frequently use my IDE's debugger mode to see the call stack. A visual form of this could be useful. The only problem is I wouldn't want to be too reliant on it to debug my code because I won't always have a tool like that to debug my code. But for learning purposes I definitely see

the use. Some kind of animation or 3D render of the stack growing and shrinking. 3D is better. I'd want it to tell me if my recursive code has any errors so I can see where it went wrong.

2) Would having a visual output rather than just a text output, encourage you to be more engaged when learning recursion?

If yes, what kind of visualizations would you like to see? What would you like to see/include in the visualizations?

If no, why do you prefer not to have a visual output? What would make you more engaged or have fun in learning recursion?

Yes, snowball/snowman increasing and decreasing in size.

Anything visual would make it more fun. Some visualization about bunnies going in and out.

Yes, image gives a different context/light. Difficult to come up with a visualization

Yes. Couldn't come up with a visualization.

Yes, hard to come up with a visualization.

Yes, couldn't think of a visualization.

Yes, more engaged. More encouraged to learn. Visualization - square over other square. Animal eating smaller animals. Animals come out of animal mouth.

Yes, something showing when each recursive call happens. So a stack of plates or blocks.

No, because you wouldn't have a visual output usually in a testing environment so it may become something you rely on too much.

Yes, like showing the items in the tower of hanoi moving.

3) Would you consider a visual coding user interface (such as a drag and drop implementation) less daunting/intimidating than regular text-based coding (such as a standard IDE) when tackling a new coding problem/concept? Why do you consider it more/less intimidating?

Do you think by first coding using a visual interface and then transition into using textual coding be good?

Less daunting. (Learner does not know what is visual coding)
But, visual coding will not be helpful.
Transition from visual coding to textual coding will be cool.

Both are equal. Typing is relevant. Transition will be good.

Easier visual. Text is fine. Transition will be good.

Visual is easier. Good mix to first use visual than text.

Easier, to learn and visualize. Harder to make mistakes with visual coding. Good to have transition from visual to text.

Yes, do not know about for growth/development of learning recursion. Testing is done with textual in theory and tests. Good for going into coding for visual, easier to code, early levels use visual coding.

Text is alright. First visual, to transitioning into text. Visual coding less mistakes - learn commands for textual coding is harder.

Text Better. You know what you are doing and understand the underlying mechanics more.

Visual is better. Transition from text to visual.

4) What do you find difficult about recursion?

Concept difficult to grasp.
Lecturers fail to explain the concept in an understandable way for the learner.
Concept needs time. Needs to be gradually taught.

Whole thing. Too fast taught - not progressive. Need more time. Too little time and pressurized.

Not taught in an easy to understand way. It's not intuitive like loops.

Didn't understand. Hard to process the concept. Needed a visual explanation of how it works - animation.

Lectures did not explain it in a simple way. Need to be giving a whole lot of exercises like converting a for loop to recursion. Learner thought they were given enough time to learn.

Difficult to visualize what's happening. Have to think of a lot of different parts like what the base case is and how the problem will get smaller with each iteration.

Don't know where recursion ends - base case. Don't know how it ends. Concept was a blur. It is new work, it is daunting, better to teach recursion early.

Loops make you lazy and became the easy way of thinking about iteration. It was difficult to switch from thinking in loops to thinking recursively.

Time as in solving question - need more time on assignments. Need more time for concept to seep in.

It's hard to compare to anything in real life.

Taught in a really awful way where it was as though the students knew what she was talking about and if it was taught in a more visual way then it would be good

Just the concept, not natural to relate to - hard to relate. Something that takes time and the seed (recursion) needs to be planted earlier.

Raw Interview Feedback:

<https://drive.google.com/file/d/1ahKWlrxWxM2PvYH87jenxK25KyGP6SZc/view?usp=sharing>