| **SI SESSION PLAN** | | SI Leader: | Audrey Fuller | Session Date: | 3/2/23 | | --- | --- | --- | --- | | Week #: | 7 | Session Letter: | B | | Course & Section: | GCIS-123.4 | Course Instructor: | Professor Audi | | Planning Date: | 2/27/23 | Planning Time: | 10:27am | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**Beginning reminders:**

☑ Is the room set up in a way conducive to collaborative learning?

☑ Is the agenda posted to the board for participants to see?

☑ Do you have your attendance sheet up to record your attendance?

☑ Do you have any other documents/resources up and ready to go for your session?

If you are all set with the reminders, then go have fun and good luck!

**Is there a study strategy you want to focus on? (If so, what is it? Otherwise, leave blank.)**

**Main concepts student should feel more comfortable with:**

Recursion

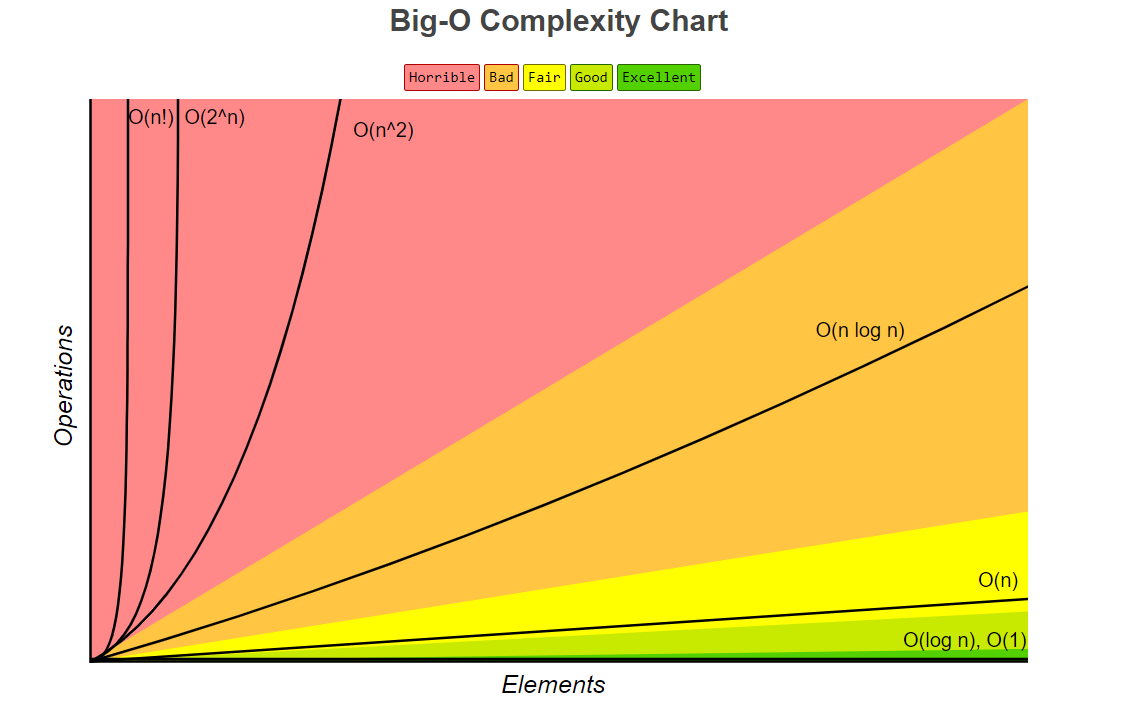
* A function that calls itself, can be used as an alternative to loops
* Base Case: part of the function that determines when to stop and does not make a recursive call (stops infinite loops)
* Recursive case: where it calls itself (handles remaining work)

The Call Stack

* Made up of Stack Frames (new one created every time a function is run)
* Most recent at top
* When a function returns, top is popped off
* Depth is the # of stack frames created

Time Complexity

* number of operations that the algorithm must perform
* Vs. Space Complexity: how much memory the algorithm needs to use
* Big O-notation



Binary Search

* If input is **Already Sorted**
  + Look at the midpoint of an array, if midpoint == target return index.
  + Else if the midpoint > target, the target must be to the left of the midpoint, so search there.
  + Else if midpoint < target, the target must be to the right of the midpoint, so search there.
* Divide & Conquer Algorithm
* Time complexity: O(log2 n)

| **Activity\*** | **Process to use** | **Time** | **After Session Thoughts** |
| --- | --- | --- | --- |
| **Opener:**  **Tier List Making** | Create a tier list of ice cream flavors at <https://tiermaker.com/create/ice-cream-flavors-48482>. If multiple people disagree on the placement, take the average of places. | 10 | | ☹ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ☺ | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ |   Took a bit longer than expected, but that was mostly because the students and I got sidetracked talking about WiCHacks this weekend. Still was a very fun activity, and I might try to make a content-oriented version of it for the future |
| **Vocab Brain Dump** | Go over vocab and concepts introduced in the last lecture (see main concepts above) in a whiteboard brain dump. I’ll write key terms on the board ahead of time, and students will each use a whiteboard marker to write anything they remember about it on the board around it. If they aren't thinking of anything to write, I’ll prompt them with questions to see what they remember from the lecture. | 20 | | ☹ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ☺ | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ |   I wish I remembered to take a picture of this activity before erasing the board, but it went well. Saige didn’t attend the last lecture so had a bit of a harder time writing down new material, but with a bit of prompting and wait time she was able to recall a bit about recursion to write on the board. |
| **Recursive Code-along** | Complete recursion coding activities as a group (each student being asked to contribute a line), including creating a recursive exponent-calculating function and a recursive string word recursive function.  <https://github.com/alf9310/GCIS-123-4-SI-Sessions/blob/main/week07/recursion.py> | 27 | | ☹ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ☺ | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ☐ | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ |   The exponential function went great, with each student contributing different pieces like the base case and recursive call. I also was able to reframe the recursive aspect for them by writing it out like an unrolling math equation on the board which helped them understand it. Unfortunately I didn’t have time to do a code-along for the word reversal function, so instead I gave them the completed code and asked them to try and explain it line-by-line. |
| **Closer: Candy Restock! What type would people like to see?** |  | 3 | | ☹ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ☺ | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ |   Saige wants more sweet tarts and Islombek requested some more granola bars (which have been pretty popular during sessions). |

*\*See the* [*Activity Database*](https://docs.google.com/spreadsheets/d/1Oc6uAX2Uaq2Ym6M1FQjivRI_ryA_T9k1AcEKi__3Ml4/edit?usp=sharing) *and* [*SI Share*](https://drive.google.com/drive/folders/1WKkkRXpRW6_OVdc4eFVgAkDRt7y8E_VT?usp=sharing) *for ideas.*

**Ending reminders:**

☑ Did you mark down attendance on your attendance sheet?

☑ Did you remind everyone of the next session and any upcoming tests or quizzes or due dates?

☑ Did you fill in the after session thoughts?

**Optional Notes and Comments:**

**Bi-Weekly Question:** How do you balance your responsibilities as an SI Leader and as a student?

I balance my SI Leader responsibilities and my student responsibilities mainly by getting my SI work (such as session plans, emails, discord messages, post-session thoughts & submissions, ect.) done during a time when I don't work on school work, while having breakfast after my SI class. This creates a solid block of time for me to just focus on SI, especially since the lecture material is still fresh in my mind.