|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SI SESSION PLAN** | |  |  |  |  | | --- | --- | --- | --- | | SI Leader: | Alex Iacob | Session Date: | 3/6/23 | | Week #: | 8 | Session Letter: | A | | Course & Section: | CSCI 141 Section 2 | Course Instructor: | Polak | | Planning Date: | 3/6/23 | Planning Time: | 3:15 – 3:30 | |

**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:**

Merge sort

Quick sort

Insertion sort

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity\*** | **Process to use** | **Time** | **After Session Thoughts** |
| **Opener:**  Wordle | It was wordle. | 3-5 | Forgot to get screenshot of the wordle   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ☹ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ☺ | |  |  |  |  |  |  |  |  |  |  | |
| **Review on sorting algos and use cards** | Have students split themselves up into groups and perform the sorting algorithms using cards that I dealt out. It is important to know what the algorithms do in each step  Insertion sort:  Pointer starts at beginning of list, checks value to its left, if left value is greater than the current card pointer, swap the cards, repeat this process until list is sorted. It is the slowest sorting algorithm we learn about.  Quick sort:  Pick a pivoting value from the list at random, make 3 new lists of cards, one which holds the values less than the pivoting value, one which holds the values equal to the pivoting value, and one with holds values greater than the pivoting value. Repeat this algorithm for each list until the original list is sorted. This is a pretty good algorithm.  Merge sort:  Split the list into two lists, then split those two lists into two more lists each. Repeat this splitting until there are lists of length two or one. Build the sorted list by progressively merging the list back together via individual comparisons. | 35-40 | People were kinda confused about everything   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ☹ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ☺ | |  |  |  |  |  |  |  |  |  |  | |
| **Practice and time complexities** | Perform Merge sort and Quick sort on array:  [8, 1, 5, 9, 2, 4, 3, 10, 6, 7]  Merge sort: All cases is O(n \* log n)  [8, 1, 5, 9, 2] [4, 3, 10, 6, 7]  [8, 1, 5] [ 9, 2] [4, 3, 10] [6, 7]  [8, 1] [5] [9] [2] [4, 3] [10] [6] [7]  [8] [1] [5] [9] [2] [4] [3] [10] [6] [7]  [1, 5, 8] [2, 9] [3, 4, 10] [6, 7]  [1, 2, 5, 8, 9] [3, 4, 6, 7, 10]  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  Quick sort: Best – O(n) : Average – O(n \* log n) : Worst – O(n ^2)  [1, 5, 2, 4, 3, 6] [7] [8, 9, 10]  [1, 5, 2, 4, 3] [6] [7] [8, 9] [10]  [1, 5, 2, 4, 3] [6] [7] [8, 9] [10]  [1, 2] [3] [5, 4] [6] [7] [8] [9] [10]  [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | 15 - 20 | Everyone had a bit of trouble working this out even though we did the same thing with playing cards under 10 minutes ago, but we got through it.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ☹ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ☺ | |  |  |  |  |  |  |  |  |  |  | |
| **Closer: Open** | Ask everyone what topics they want me to cover for the remaining time. | remaining | Ran out of time at the end, didn’t get to this.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ☹ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ☺ | |  |  |  |  |  |  |  |  |  |  | |

*\*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:** Do you find you now have regulars at your sessions? If you don’t mind sharing, tell us about some of them! If you don’t have any regulars, how can you encourage students to attend regularly?

I have a few regulars that are showing up now, which is pretty nice. Most of them are students that are repeating the course or are non-CS majors. One of them was the student that I talked about during Friday training who broke down fully. He seems to be doing better now. I have another regular who is trying to make her own major, which is pretty cool, she mentioned that she is doing CS1 entirely out of her own free will. My other two “kinda” regulars have started coming recently and they are more or less at the top of the class.

I also have a different set of students that come to my A and B sessions, so my regulars differ a bit.