|  |
| --- |
| **Data Structure Workouts** |
| 1. Learn about Bubble Sort, Insertion Sort, Selection Sort, Quick sort and Merge sort. Complete at least three sample workouts in each of them. 2. Learn the concept of Stack and Queue. Complete at least three sample workouts in each of them & do at least 3 problems from any competitive coding websites (Hacker Rank, Code Chef, Leet code, Algo Expert, etc.)    1. PUSH, POP and Display elements in Stack    2. Enqueue, Dequeue and Display elements in Queue 3. Learn the concepts of Hash Table. Complete at least 3 sample workouts. 4. Learn about the Applications of all structures you covered this week |
| *Write a short description about this task*  This task is very interesting to me because before this week I only knew about selection sort only , but after studying all sorting methods now I am capable of doing a sorting task with different algorithms , this is a really helpful task to me. I personally like Merge sort because  The complexity of merge sort is O(n log n) in the worst case.  Bubble Sort repeatedly swaps adjacent elements if they are in the wrong order, with a time complexity of O(n^2) in the worst case.  Insertion Sort iterates through an array, inserting each element into its correct position among the elements that have already been sorted, with a time complexity of O(n^2) in the worst case.  Selection Sort repeatedly selects the smallest remaining element and swaps it with the current element being sorted, with a time complexity of O(n^2) in the worst case.  Quick Sort selects a pivot element, partitions the array around it , and recursively sorts the resulting sub-arrays , with a time complexity of O(n\* log n) in the worst case.  Merge Sort divides the array into two halves, recursively sorts them, and then merges the sorted halves, with a time complexity of O(n\*log n) in the worst case.  *Link to the folder containing code and screenshot of the output* |
| *Write a short description about this task*  *This task helps me to understand two fundamental data structure in cs, that are used to store data and organize data. Both stack and queue are abstract data types that can be implemented using various programming languages. A stack is a linear data structure that follows the Last- in- First -Out(LIFO) principle. The two primary operations of a stack are push() and pop().I studied the push() and pop() very well through this task . Queue is also a linear data structure but follows the First-first-out (Fifo)principle . The two primary operation of queue are enqueue and dequeue . I studied these two operation also very well.*  *Link to the folder containing code and screenshot of the output* |
| *Write a short description about this task*  *This task helps me to study the hashtable data structure. At first time this is very difficult to me. Then I went through the code then i understand the logic . First time i heard about hash function and all i just thought what is going on in hash table. Then i followed a youtube channel code evolution and tute box . It helps me to understand the whole data Structure. His playlist speciality is he will show a slide how data structure works. A hash table is a data structure that stores data in an associative array format, where each data stored like a key and value pair. Hash table provide efficient search, insertion, and deletion operations making them an essential data structure .*  *Link to the folder containing code and screenshot of the output* |
| *Write a short description about this task*  *This task helps me to understand the data structure I covered in this week where it is applicable . This understanding makes me more Perceive about all structures.*  *Stacks can be used in function calls, expression evaluation and undo/redo functionalities, then Queue can be used in job scheduling, network buffering etc and at last hashtable can be used at implementing symbol tables, caching, counting occurrences.*  *While knowing about where these data structures are used, its new knowledge to me.* |