

172164-group B-DBIT 1301 CAT2

PART1

B: my chosen data structure is a Linked list and stack: and my chosen problem is managing a task scheduler.

Ok so the linked list is dynamic in nature hence this allows it to be able to allocate memory whenever it is required. Also, insertion and deletion can be easily implemented and its main advantage is that it reduces the access time hence managing tasks becomes easier. Stack is the most suitable data structure for handling or managing tasks, especially in LIFO order because its elements are added and removed from the same end.

C: linked list has a fast insertion and deletion meaning the stacks allow fast insertion and deletion of elements but it also has limited access where stacks only allow access to the top element and stacks are used for systematic memory management, used in many virtual machines like JVM, are more secure and reliable as they don't get corrupted easily but it also has its disadvantages like limited memory size where the stack is very limited and also random access is not possible also creating too many objects on a stack can increase the risk of stack overflow.

PART2

a)

ii) Big O: This is because copying the top 10 elements from the sorted array is all that is necessary and consumes time.

Big omega: This is because copying the top 10 elements from the sorted array is all that is required, and it takes time.

Big theta: This is because copying the top 10 elements from the sorted array is all that is required, and it takes time.

ii) The sorting step, with a time complexity of $\Theta(n^2)$, dominates the algorithm's overall time complexity. The overall time complexity is not significantly affected by the step of choosing the top 10 transactions.