**Classroom Activities for this Lesson**

**ACTIVITY 1**

The purpose of this activity is to make students realize the limitations of using an array & the need of a new & better data structure for storage purpose.

Two activities will be performed in class.

1- Define & array of size 10 & fill it with some data. Then ask the students that if there is a need to insert one more element then what can be done? Of course a new array with bigger size will be declared, all the data will be copied in it & new data will be added after that.

2- What if three elements are already added in array (say apple, banana & orange) keeping alphabetical order in mind. Now you want to insert another element "Mango" in it. In case of an array there is a need to move the entire elements one index further that comes after mango and place mango on its place.

Moving further with these two activities make students realize the need of a structure that can increase or decrease its size dynamically & there is always a room to insert data anywhere without the need of disturbing the others.

**ACTIVITY 2**

In this activity the teacher has to make students realize that how a linked list can be more useful in certain scenario by counting the steps that will be taken by the same two examples in the previous activity.

1- To evaluate the first clause i.e. the dynamic increase in size, take a bench where three students are sitting. (Denoting an array). What if another student comes and wants to sit with them. A new bigger bench will be brought, all three will shift to the new bench & the $4^{th}$ will sit with them. But in case of linked list the students will be sitting on separate chairs & the next coming student just needs to bring his chair to join them. Count the steps in both implementations.

2- To evaluate the second clause i.e. insertion in array let some students stand next to each other (Denoting and array). If new data needs to be inserted at a certain point, all the data after the desired index needs to be shifted one location further. But in case of a linked list consider the chain of boys holding each other's hand to form links. If new data needs to be inserted then only the desired location will break its link & hold the hands of the upcoming boy or node. Let them count the steps here as well.

**Activity 3**

This activity will build upon the scenario presented in segment 3, and make students realize that the "list" that they are maintaining is starting to exhibit the properties of another data structure.

Suppose that we had to bring 3 different books for 3 friends that met you on the way to the book store: we would obviously remember their requested books in the form of a list in our mind, and would also maintain that list in the order in which we met our friends.

Hence when we will ask for those books at the store, we would be going one by one following the list, and let's assume we place the books on top of each other as the shopkeeper hands them to us. Now we would have a pile of books.

When we would be taking those books back, we would be backtracking on the path we came from, and the friend that met us last, would come first, and the one that was first would now meet us last. Interestingly, this is the exact same order in which our books are piled up.

Now, as we meet our friends along the way, we would keep handing them the book at the top of our pile one by one and it would automatically be in order.

Let's suppose we don't backtrack exactly the way we came, and meet all of our friends in a random order to hand them their books. We would have to stifle through the pile in order to find their desired book and then hand it to them.

We just had 3 books in this case. Imagine if we had 10. Wouldn't it be a hassle to pull out a book from between a big pile? And what if we pulled out a book from the bottom and the pile fell? This is called "Random access", and we can see that in our situation that is not a good idea.

So obviously, we won't be pulling books from the bottom if this is the case, we would one by one, remove the top book, and keep piling them up on top of each other (notice the reverse order), and when we come to our desired book, we would hand it to our friend and put the books back in their order, on top of each other.

Here, we have interestingly encountered another data structure, whose properties have been demonstrated through the example presented.

Introduce students with the concept of stack & resume the lesson video.

The activity is to make students realize the uses of stack.

**ACTIVITY 4**

1- Write something on the board or type something on the keyboard. Then erase it. Ask the student that which pattern is followed here. They'll notice that the last one to be written will be the first one to be erased.

2- Repeat the same thing with opening the folders in windows. When backspace is pressed the latest open folder will be the first one to be closed. Same thing can be done with clicking the links on webpages.

Give them some time to think about more examples in computer science & resume the video.

**ACTIVITY 5**

This activity aims to identify the flaw of stack if used in a scenario where a queue is needed.

Let them play an act of shopkeeper & customer.

One student may become shopkeeper & the others will be customers. After every 10 seconds a new customer will come & add himself in a stack. The shopkeeper serves the customer once every 15 seconds using the LIFO fashion.
Make them realize that in this fashion the one who came first may never get his turn & will be waiting forever.

Introduce with the concept of queue & resume the video.

**ACTIVITY 6**

This activity will be a short exercise & may be held as a voting session. You may ask the class that out of the three data structure list, stacks & queue, which will be the best one ? ask every students to share his views.
At the end make them realize that no data structure is best for everything. It depends what you need & structure will be used accordingly. Summarize this discussion & resume the video.