

You.

Let's talk about queue and stack, two very similar data structures that are built either on top of a vector or on top of a deck type of data structure. In fact, these data structures are really container adapters, both of them. So the difference between queue and a stack is queue is when you go to a store, you might stand in a queue. It's going to be first come, first served. So it's, it's a first in, first out type of data structures.

If you get into the queue early, you get serviced earlier. Stack, on the other other hand, is a last in, first out kind of data structure. You may want to think of it as a stack of plates. You put things on top and you take things from the top. So let's see how these two data structures will get used.

So first let's look at Q operation operations.

We can have a queue of any data type. So we are creating a queue with 10, 20 and 30. We can look at our queue, it's got a size of three and while it's not empty, we can go through our elements. So the front element of the queue, the front of the queue is now going to be 10. Next one is going to be 20 and the one after that 30.

The front accesses the front element of the queue, but doesn't actually remove it. To remove the element, we use queue and pop as part of it.

So our next operation, next data structure is the stack. Once again, we create a stack of integers. We have 10, 20 and 30 as part of our stack, okay?

So we can look at the size of our stack and the size of our stack is going to be three. And while our stack is not empty, we can take things from the top of the stack. So we take the top element of the stack and remove an element. We remove it using pop as the operation. So these are Q and stack and we'll get to see what kind of problems they get used in later.