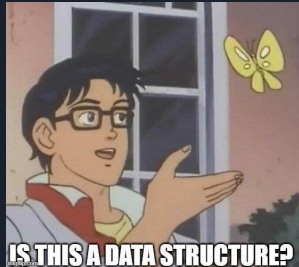


A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one in front of the green one.

Array Manipulation

Presented by Alexander Mak

What are data structures and algorithms anyways...?



Array Metaphor

Imagine that you are a librarian who is responsible for organizing a vast collection of books.

You can imagine an array as a shelf on the bookshelf. Each slot on the shelf holds a book. Just like how you can easily find a book on specific shelf if you know what slot it is in, you can find a specific element in an array if you know its index.

In our library example, arrays would be especially useful for organizing books by categories. You could have an array for history books, science books, and so on. This would make it easy for people to find the books they need, and for you to keep track of your collection.

So, just as a librarian uses shelves and slots to organize books, programmers use arrays to organize collections of values.





Arrays allocated in RAM

Array

1	3	5
---	---	---

RAM

Value			1	3	5				
Address			\$0	\$4	\$8				



Arrays allocated in RAM

Array

1	3	5
---	---	---

- Arrays are contiguous in memory
- Traditional arrays are FIXED in size
 - This is a “static array”
- Many modern languages such as Python and Javascript use “dynamic arrays” that will resize themselves as needed

RAM

Value			1	3	5				
Address			\$0	\$4	\$8				

How does a dynamic array work?

Array

1	3	5
---	---	---

RAM

Value	1	3	5		
Address	\$0	\$4	\$8		

Value	1	3	5							
Address	\$0	\$4	\$8							

- If half an array is filled, the allocated space for an array is doubled
- Since arrays must be contiguous and we cannot guarantee the memory next to the current array is available, we must reassign the entire array.
- This is $O(n)$, but it happens so infrequently that we can say the average case is $O(1)$.
 - This is called **amortization**



Let's recap: Arrays

- Arrays must be contiguous in memory
- Static arrays have a fixed size
- Dynamic arrays solve our space problem if we fill up the array with values and are the default for many languages such as Javascript or Python
- Whenever a dynamic array needs to be resized when adding new elements, it is $O(n)$. However, the amortized time complexity is $O(1)$.

Operations	Big-O Time
Read/ Write ith element	$O(1)$
Insert / Remove End	$O(1)$
Insert Middle or Beginning	$O(n)$
Remove Middle or Beginning	$O(n)$



Traversing Arrays

- Arrays are typically traversed using loops
 - for loops are common but other loops such as while loops can make sense in certain scenarios as well
- When traversing multi-dimensional arrays (aka nested arrays), we want to traverse outside-in
- Example:

```
matrix = [  
    [1, 2, 3, 4, 5],  
    [6, 7, 8, 9, 10],  
    [11, 12, 13, 14, 15]  
]
```

- In Python, range takes in 3 parameters.
- range(start, exclusive_end, increment)
- Start and increment are optional and are set to 0 and 1 by default.

```
for r in range(len(matrix)):  
    for c in range(len(matrix[0])):  
        print(matrix[r][c])
```


Questions?



Let's practice!

https://github.com/Dijkstra-LLC/dsa_live_pro/tree/main/W01D01/classwork