Static and Dynamic Arrays

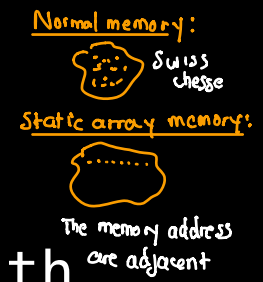
William Fiset

Outline

- Discussion and examples about Arrays
  - What is an Array?
  - When and where is a Array used?
  - Complexity
  - Static array usage example
- Dynamic Array implementation details
- Code Implementation

Discussion and examples

What is a static Array?



A static array is a fixed length container containing n elements **indexable** from the range [0, n-1].

Q: What is meant by being ‘indexable’?

A: This means that each slot/index in the array can be referenced with a number.

When and where is a static Array used?

1) Storing and accessing sequential data

2) Temporarily storing objects

3) Used by IO routines as buffers

4) Lookup tables and inverse lookup tables

5) Can be used to return multiple values from a function

6) Used in dynamic programming to cache answers to subproblems

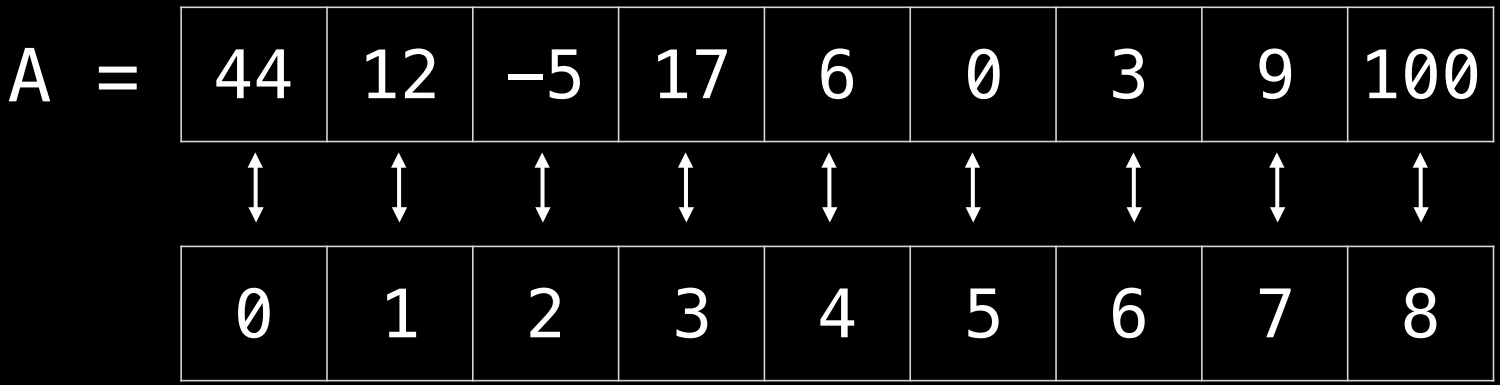
Complexity

O(1) bc of the property that arrays are indexible

up to O(n) bc we may traverse all elements

	Static Array	Dynamic Array
Access	O(1)	O(1)
Search	O(n)	O(n)
Insertion	N/A (Array is of fixed size)	potentially move all to right and O(n) recopy elements to static
Appending	N/A	O(1)
Deletion	N/A	O(n)

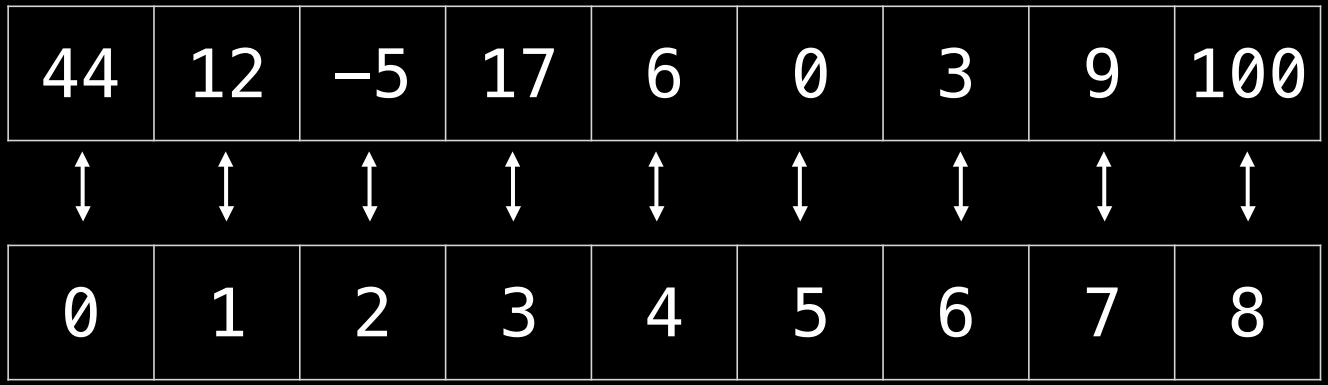
Static Array



Elements in *A* are referenced by their index. There is no other way to access elements in an array. Array indexing is zero-based, meaning the first element is found in position zero.

Static Array

A =



A[0] = 44

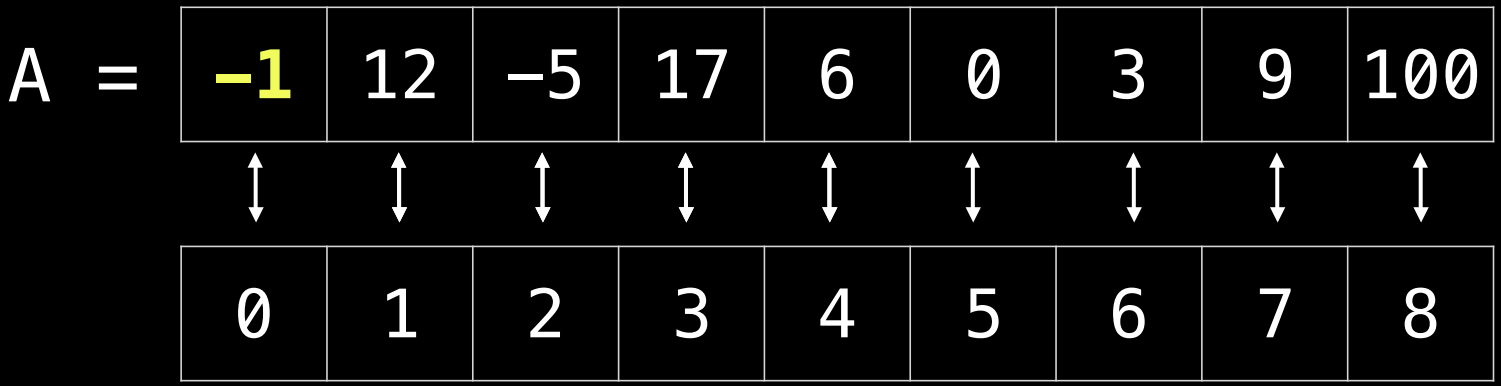
A[1] = 12

A[4] = 6

A[7] = 9

A[9] => index out of bounds!

Static Array



A[0] = 44

A[1] = 12

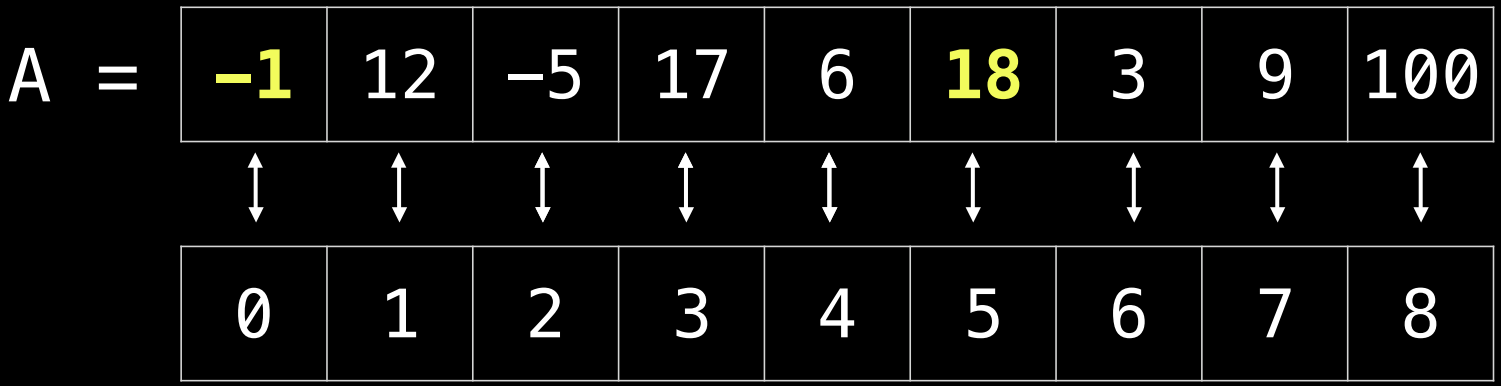
A[4] = 6

A[7] = 9

A[9] => index out of bounds!

A[0] := -1

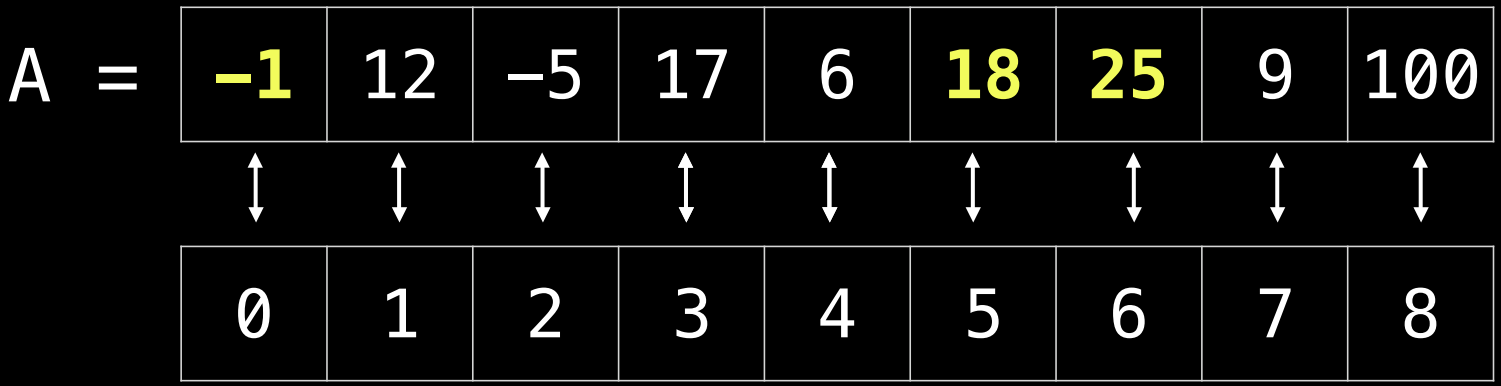
Static Array




```
A[0] = 44
A[1] = 12
A[4] = 6
A[7] = 9
A[9] => index out of bounds!
```

```
A[0] := -1
A[5] := 18
```

Static Array




```
A[0] = 44
A[1] = 12
A[4] = 6
A[7] = 9
```

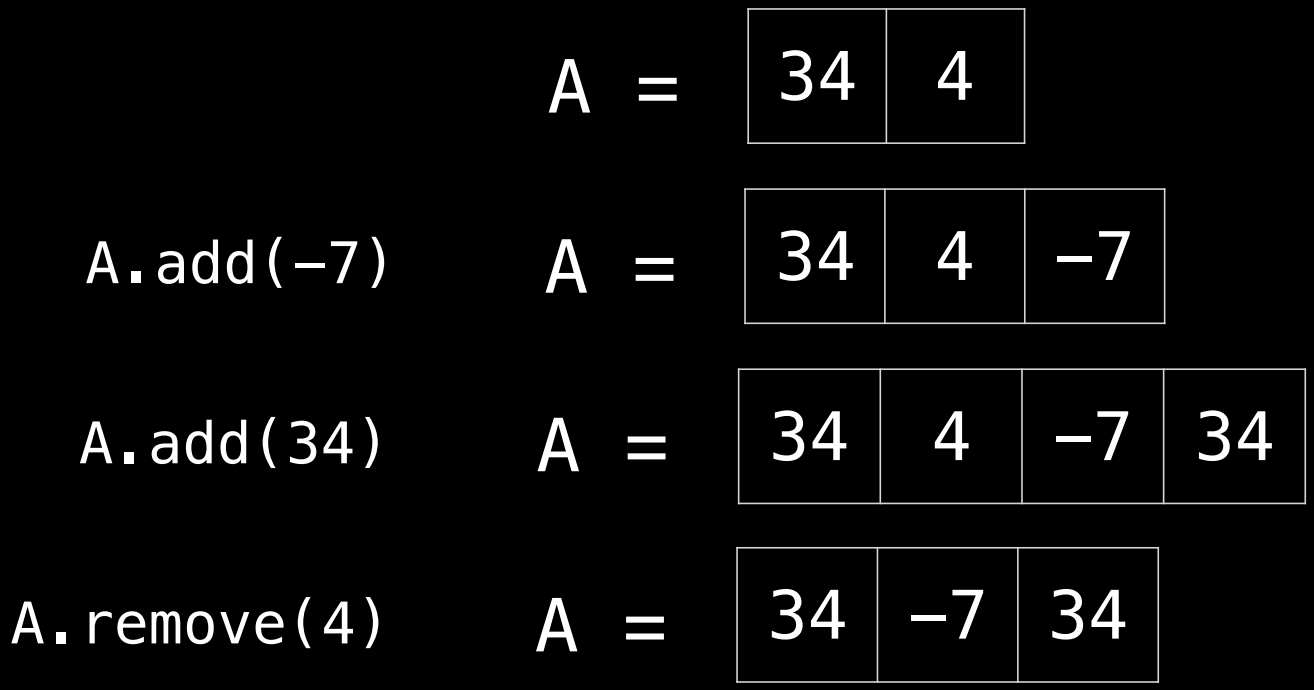
```
A[0] := -1
A[5] := 18
A[6] := 25
```

A[9] => index out of bounds!

Operations on Dynamic Arrays

Dynamic Array

The dynamic array can **grow** and **shrink** in size.



Dynamic Array

Q: How can we implement a dynamic array?

A: One way is to use a static array!

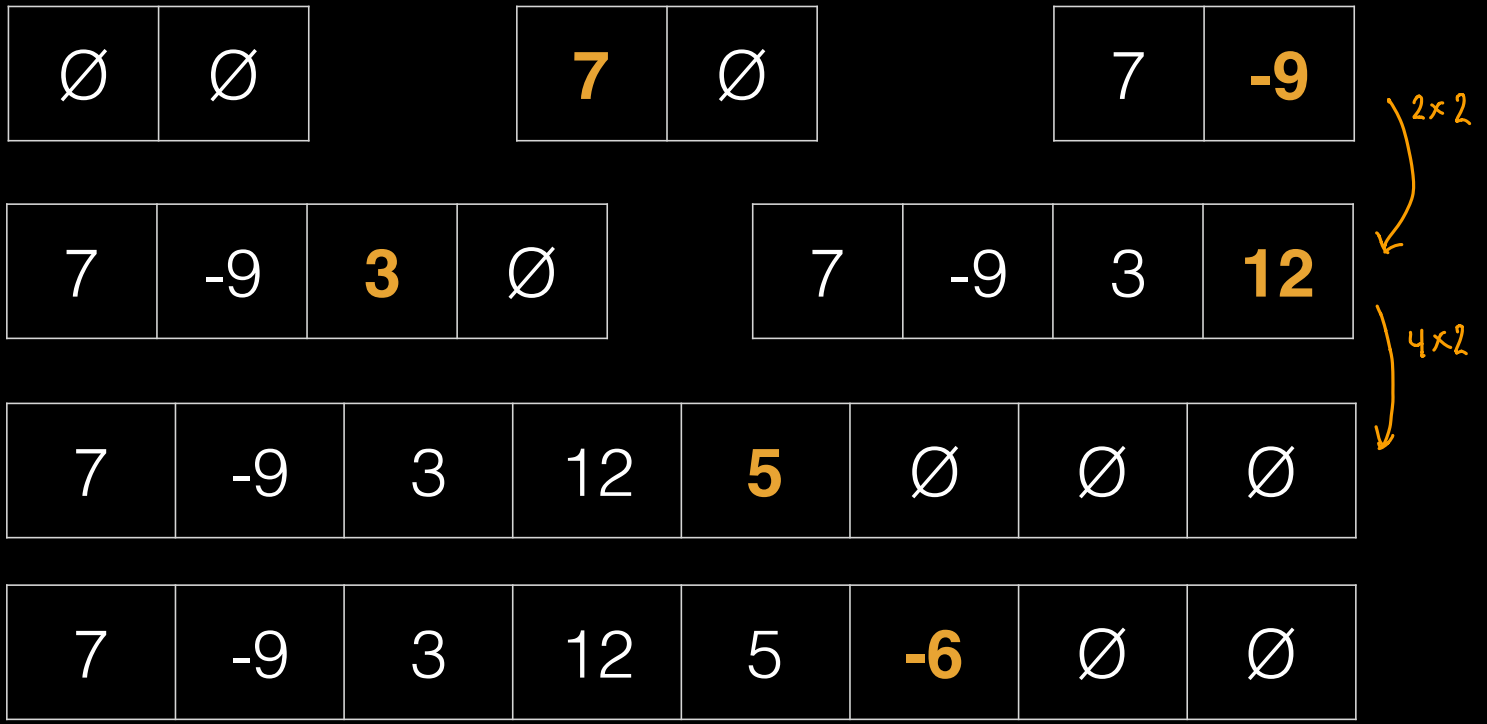
1) Create a static array with an initial capacity.

2) Add elements to the underlying static array, keeping track of the number of elements.

3) If adding another element will exceed the capacity, then create a new static array with twice the capacity and copy the original elements into it.

Dynamic Array

Suppose we create a dynamic array with an initial capacity of two and then begin adding elements to it.



1. What is the difference between array and dynamic array?
2. What is the corresponding built-in data structure of array and dynamic array in your frequently-used language?
3. How to perform basic operations (initialization, data access, modification, iteration, sort, etc) in an array?
4. How to perform basic operations (initialization, data access, modification, iteration, sort, addition, deletion, etc) in a dynamic array?

724. Find Pivot Index

Given an array of integers `nums`, write a method that returns the "pivot" index of this array.

We define the pivot index as the index where the sum of all the numbers to the left of the index is equal to the sum of all the numbers to the right of the index.

If no such index exists, we should return -1. If there are multiple pivot indexes, you should return the left-most pivot index.

Example 1:

```
Input: nums = [1,7,3,6,5,6]
Output: 3
Explanation:
The sum of the numbers to the left of index 3 (nums[3] = 6) is equal to the sum of numbers to the right
Also, 3 is the first index where this occurs.
```

Example 2:

```
Input: nums = [1,2,3]
Output: -1
Explanation:
There is no index that satisfies the conditions in the problem statement.
```

Constraints:

- The length of `nums` will be in the range `[0, 10000]`.
- Each element `nums[i]` will be an integer in the range `[-1000, 1000]`.

JavaScript

```javascript
/**
 * @param {number[]} nums
 * @return {number}
 */
var pivotIndex = function(nums) {
    if (nums.length === 0) return -1;
    if (nums.lengt === 1) return 0;

    const sum = nums.reduce((acc, cur) => acc + cur);

    let leftSum = 0;

    for(i = 0; i<nums.length; i++) {
        if (leftSum === sum - nums[i] - leftSum) {
            return i;
        }

        leftSum += nums[i]
    }

    return -1;
};
```

Time complexity:
O(N), where N is the length of nums

Space Complexity:
O(1), the space used by leftSum and sum

nums = $[a_1, a_2, a_3, a_4, a_5]$;

Sum entire array

Subtract left of pivot

Subtract pivot

Sum of right side

a_3 is a pivot if $a_1 + a_2 = a_4 + a_5$

Key concepts

Array.reduce()

===