Algorithms and Data Structures: Array

Zeshan Fayyaz

Overview:

- Arrays hold values of the same type at adjacent memory locations
    - Concerned about two things - the index of the element as well as the element itself
- In Python, the size is dynamic and therefore we do not need to have a size defined beforehand
- We can store multiple elements of the same type with one single variable name
- We can access elements as long as we have the index, as opposed to linked lists (where we have to traverse from the head)
- Addition and removal of elements into/from the middle of the array is slow because the remaining elements need to be shifted
- The index is always less than the total number of array items
- Python has a special module for creating arrays, called "array" which must be imported before using

In Python, arrays are different from lists and are initialized as:

```
arrayName = array.array(type code for data type, [item1, item2, etc.])

import array
balance = array.array('i', [300, 200, 100])
print(balance)
```

- To access a specific item in the array → `arrayName[indexNum]`
- To insert at a specific index → `arrayName.insert(index, value)`
- To delete one item from an array by value → `arrayName.remove(value)`
- We can search for an item in an array based on its value → `arrayName.index(value)`
- Update values within the array (replace an existing value) → `arrayName.update(index, value)`

Terminology:

- A subarray is a range of adjacent values within an array.
    - For example, given the array [5, 7, 7, 8, 3, 4] a subarray can be [7, 8, 3] while [5, 8, 4] is not
- A subsequence is a sequence that can be derived from the original array by deleting some or no elements without changing the order
    - For example, given the array [2, 5, 7, 3, 5] a subsequence can be [2, 3] while [3, 2] is not

Time complexities:

- Access O(1)
- Search O(n)
- Insert O(n)
- Insert at the end of the array O(1)
- Remove O(n)
- Remove at the end of the array O(n)