Exercise I

Basic Coding Problems

Array

To excel in coding interviews, a deep understanding of arrays—a crucial data structure—is essential. Engaging with a broad spectrum of array-related problems, from basic to advanced, is not just beneficial but necessary. This rigorous practice forms the bedrock of your problem-solving arsenal, equipping you to navigate the complexities of algorithmic challenges with confidence. Commit to mastering arrays; it's an imperative step in your journey towards technical proficiency and interview readiness. **Here, we will try solving some introductory problems.**

Array

- What is Array?
  - An array is a collection of items stored at contiguous memory locations.
  - The idea is to store multiple items of the same type together.
  - This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (generally denoted by the name of the array).
- Python does not have built-in support for Arrays, but Python Lists can be used instead.
- Or `numpy.array`

Array (review)

- Creating an array

```
mylist = [3, 1, 2, 1, 5, 4]
```

- Accessing an element of the array

```
mylist[3] = 7

print(mylist[1])
```

- Length of the array

```
print(len(mylist))
```

Lab 17-1 Warming Up!

Given an array `myarray` of integers, return how many of them contain an even number of digits.

Input: `myarray = [17, 462, 4, 5, 8575]`

Output: `2`

Lab 17-2 Warming Up!

Given a binary array, find the maximum number of consecutive 1s in this array.

Example

```
myarray = [1, 1, 0, 1, 1, 1, 0, 0, 1, 1]

Output: 3
```

Inserting, Removing, and Searching

Three key operations for Arrays:

1. Inserting items.
2. Removing items.
3. Searching for items.

Random int Array

- random module

```
import random
```

- Generating a random number

```
num = random.random()
```

- Generating a random int number

```
num = random.randint(0,100)
```

Random int Array

- Random int array (length 10)

```
num = []

for i and range(10):

    num.append(random.randint())
```

Random int Array

- Random int array (length 10)

```
num = random.sample(range(0, 15), 10)
```

Inserting an item in the middle

- Add a number at index 2 using **a built-in function**

```
num = random.sample(range(0, 15), 10)

num.insert(2, 777)
```

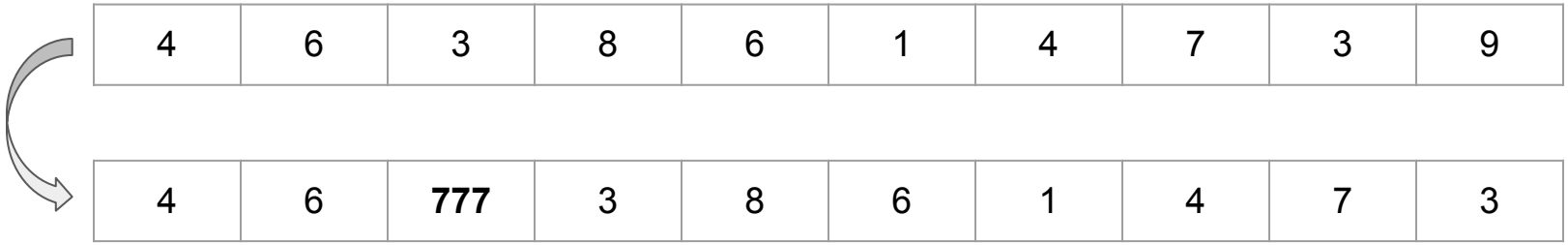
- Add a number at index 2 without using a built-in function

???

Lab 17-3

In a random integer array of length 10, insert the number 777 at index j without utilizing any built-in functions. You must operate directly on this array without allocating space for a second array and avoid using slicing. Following the insertion, assume the final element of the array is overwritten and lost.

For example, if `j` = 2



Lab 17-4

Given a fixed length array arr (size 10) of integers, duplicate each occurrence of `k`, shifting the remaining elements to the right. For example, if `k` = 2

Input:

arr = [0, 1, 3, **2**, 4, **2**, 0, 0, 0, 0]

Output:

arr = [0, 1, 3, **2**, **2**, 4, **2**, **2**, 0, 0]

Lab 17-4

Input:

arr = [3, **2**, **2**, 5, 1, **2**, 0, 7, 8, 9]

Output:

arr = [3, **2**, **2**, **2**, **2**, 5, 1, **2**, **2**, 0]

Input:

arr = [7, 4, 3, 6, 3, 9, 5, 0, 4, 4]

Output:

arr = [7, 4, 3, 6, 3, 9, 5, 0, 4, 4]

Deleting an item in the middle

- delete the number at index 2 using **a built-in function**

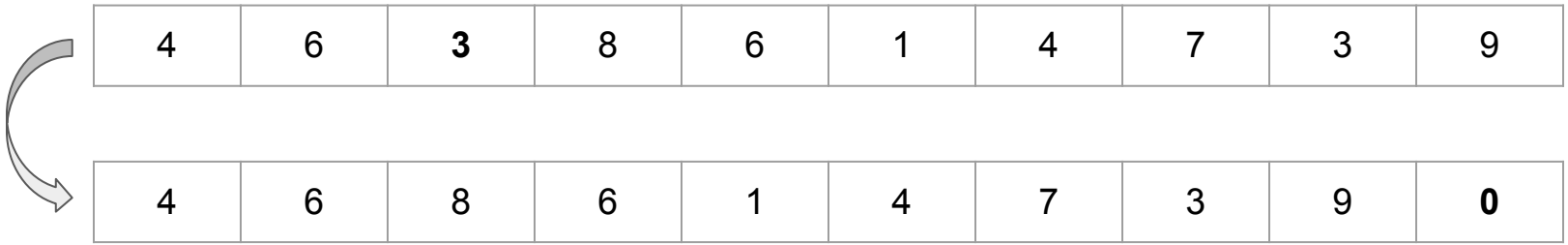
```
num = random.sample(range(0, 15), 10)

num.pop(2) # or del num[2]
```

- delete the number at index 2 without using a built-in function

???

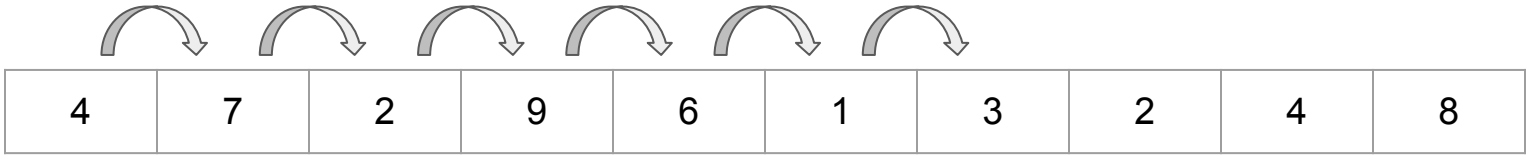
Lab 17-5

Given a random integer array (length 10), delete a number at index `j` without using a built-in function. Suppose that we put zero for the last element after deleting. For example, if `j` = 2



Searching in an array (Linear Search)

We continue checking elements until we find the element we're looking for, or we reach the end of the Array. This technique for finding an element by checking through all elements one by one is known as the linear search algorithm. For example, if we are looking for 3,



Lab 17-6

Given an array arr of integers, check if there exists two integers N and M such that N is the double of M (i.e. N = 2 * M).

More formally check if there exists two indices i and j such that :

i != j

0 <= i, j < len(arr)

arr[i] == 2 * arr[j]

Lab 17-6

For example,

input:

arr = [10, 2, 5, 3]

Output:

True

input:

arr = [7, 1, 14, 11]

Output:

True

input:

arr = [3, 1, 7, 11]

Output:

False