Implementation of Binary Search

In this notebook we will just implement two versions of a simple binary search. View the video lecture for a full breakdown!

Binary Search

In [4]:
```
def binary_search(arr,ele):

    # First and last index values
    first = 0
    last = len(arr) - 1

    found = False


    while first <= last and not found:

        mid = (first+last)//2

        # Match found
        if arr[mid] == ele:
            found = True

        # Set new midpoints up or down depending on comparison
        else:
            # Set down
            if ele < arr[mid]:
                last = mid -1
            # Set up
            else:
                first = mid + 1

    return found
```

In [5]:
```
# list must already be sorted!
arr = [1,2,3,4,5,6,7,8,9,10]
```

In [6]:
```
binary_search(arr,4)
```

Out[6]:
```
True
```

In [7]:
```
binary_search(arr,2.2)
```

Out[7]:
```
False
```

Recursive Version of Binary Search

In [10]:
```
def rec_bin_search(arr,ele):

    # Base Case!
    if len(arr) == 0:
        return False
```

```
    # Recursive Case
    else:

        mid = len(arr)//2

        # If match found
        if arr[mid]==ele:
            return True

        else:

            # Call again on second half
            if ele<arr[mid]:
                return rec_bin_search(arr[:mid],ele)

            # Or call on first half
            else:
                return rec_bin_search(arr[mid+1:],ele)
```

In [11]:
```
rec_bin_search(arr,3)
```

Out[11]:
```
True
```

In [12]:
```
rec_bin_search(arr,15)
```

Out[12]:
```
False
```

Good Job!