Chapter 5.5: Searching and Sorting Algorithms
Binary Search

Introduction
The binary search works by splitting a list of SORTED items into half each iteration. It always goes to the midpoint, checks to see which half the item searched for is and then continues to split the list in half each time.

Example
Given the numbers **3, 6, 8, 9, 12, 15, 18, 24, 27**
Find number **9**

3	**6**	**8**	**9**	**12**	**15**	**18**	**24**	**27**
3	**6**	**8**	**9**					
9								

Task 1:
Given the numbers **3, 15, 21, 27, 33, 39, 42, 48, 56, 60, 66, 67, 69**
Find number **21**
In the space below complete the table. Add as many rows as you need by clicking tab when you are in the last cell and If you want to merge cells like I did above you just highlight them and merge cells.

3	**15**	**21**	**27**	**33**	**39**	**42**	**48**	**56**	**60**	**66**	**67**	**69**
3	**15**	**21**	**27**	**33**	**39**	**42**						
3	**15**	**21**	**27**									
21	**27**											
21												

Task 2:
Given the numbers **3, 5, 6, 8, 9, 12, 15, 21, 23, 45, 56, 63, 69**
Guess the number by following:

- The first guess was too low
- The second guess was too high
- The third guess was high
- The fourth guess found the number.

The number was?

21

Task 3:
Before writing a line of code a student wrote down the steps to better understand how they could code a binary search. The steps are not in the correct order. Rearrange the steps so the algorithm would work. There are a couple of empty cells so you can copy and paste items out of the way

1) Enter a number
7) If middle is less than number entered then start equals middle +1
10) Inform the user that the number is not present
4) End of search items equals length of array -1
11) End of while loop
5) Middle equals (start+end) /2
3) Find the length of the array
8) If middle is greater than number entered then end equals middle -1
9) While start is less than or equal to end
6) If middle is equal to number entered tell the user and stop the loop
2) Start (of search items) equals 0

Task 4:
If you had a list of 100 items and you had to find an item in the array. What would be the worst case scenario (Maximum number of checks) for:

Linear search

100

Binary search

7

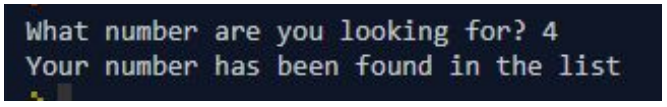
Extension

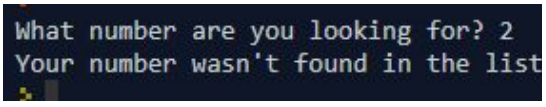
Chapter 5.5: Searching and Sorting Algorithms
Binary Search
3 of 3

Investigate how to perform a Binary search in Python and have a go at using one with some random numbers.

```
def binarySearch(myItem, myList):
  found = False
  bottom = 0
  top = len(myList) - 1
  while bottom <= top and not found:
    middle = (bottom+top) // 2
    if myList[middle] == myItem:
      found = True
    elif myList[middle] < myItem:
      bottom = middle + 1
    else:
      top = middle - 1
  return found

if 1==1:
  numberList = [1, 4, 6, 8, 12, 15, 18, 19, 24, 27, 31, 42, 43, 58]
  item = int(input("What number are you looking for?"))
  isItFound = binarySearch(item, numberList)
  if isItFound:
    print("Your number has been found in the list")
  else:
    print("Your number wasn't found in the list")
```





http://pythonschool.net/data-structures-algorithms/binary-search/