Homework 0 - Binary Search

# Instructions

In this homework, you will be writing a common search algorithm, binary search. In binary search, you are given a list (which MUST be sorted) and an element. The goal is to return the index of the element if it exists in the list, and -1 if that element does not exist (NOTE the difference between an index and an element). Linear search, which you wrote last class, will also work for this -- however, it is much slower than binary search, especially on huge lists.

First, watch this video: <https://www.youtube.com/watch?v=JQhciTuD3E8>

Then, read the below overview of binary search (in “Binary Search Explained”).

Lastly, open up “binarySearchActivity.py.” You will find challenge statement at the bottom of the file. Read each challenge one at a time, and then complete it. They will walk you through how to code up binary search. I know there is a lot of text in the challenges, but don’t get overwhelmed -- the actual code you have to write is just 8 lines (the text is just to walk you through the code you are supposed to write).

We have included “linearSearch.py”, which is an example of linear search so you have an idea of correct Python syntax. Feel free to also refer to the Python files from last class, which should demonstrate a lot of the syntax. Also feel free to look at this website, which has many examples of correct Python syntax: <https://learnxinyminutes.com/docs/python3/>

Please email us at [cmu.teknowledge@gmail.com](mailto:cmu.teknowledge@gmail.com) if you have any questions (do email us -- we love getting questions)!

# Binary Search Explained

In binary search, we are given a **sorted** list and an element to find in the list. We first check the middle element of the list. If the element we are looking for is greater than it, we ignore the whole left half of the list (which will ALL be lesser than the element we want) and look at the right half. If it is lesser, we ignore the whole right half and just look at the left half. If it is equal, we have found our element :)

Now that we have eliminated half of the list, we continue looking at the midpoint of the remaining list, until we have either found the element or we are left with a list of size 0 (which cannot be cut in half any longer). If we are left with a list of size 0, we know the element does not exist and return -1.