**Algorithm Design: Searching**

Selecting specific data.

1. Linear Search

* Most Basic Search.
* Iterates over all elements until item is found

The element to be found is called a Search Key.

* If the element is not found, it is termed a failure.

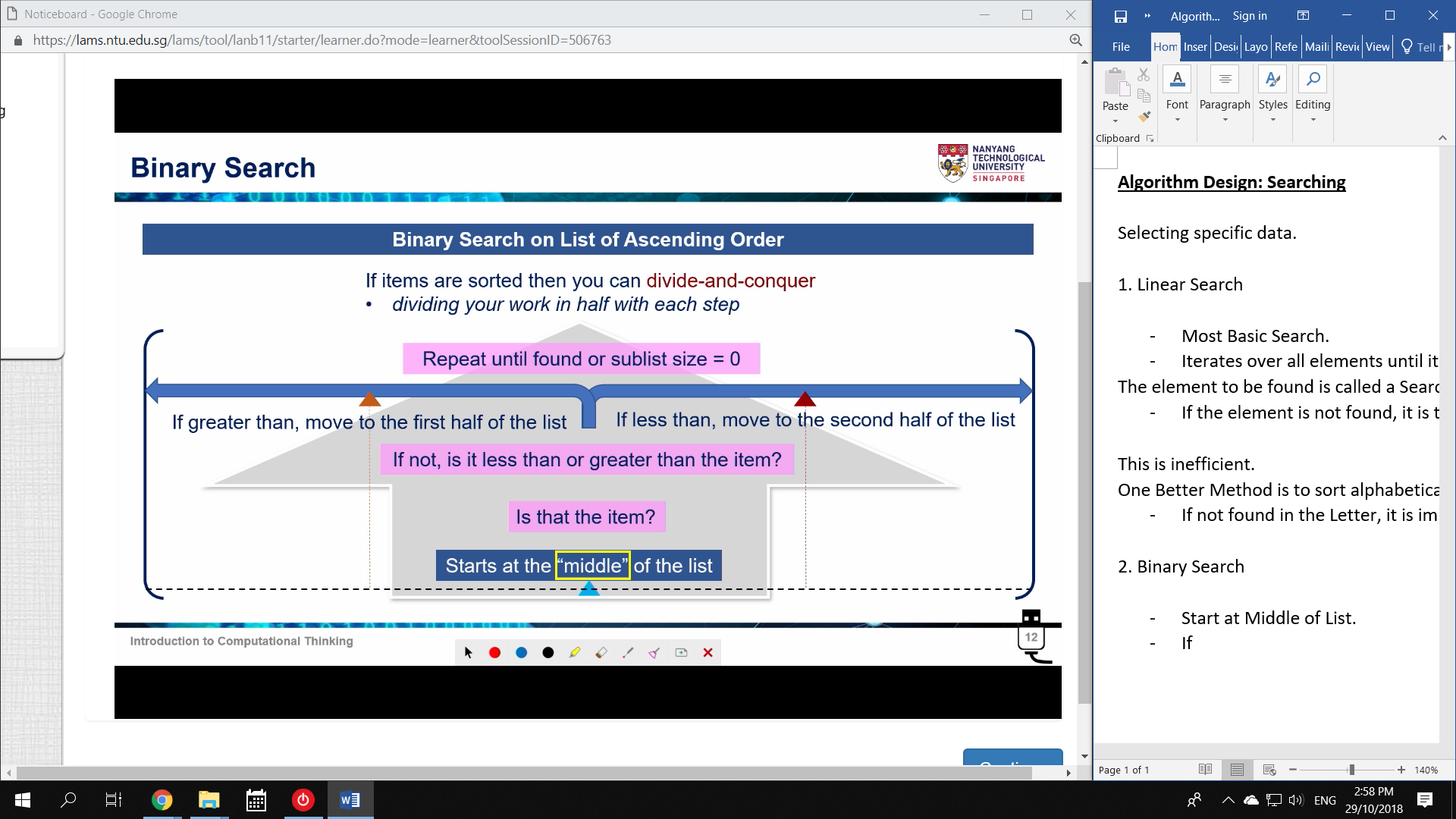
This is inefficient.

One Better Method is to sort alphabetically.

* If not found in the Letter, it is immediately termed a failure.

2. Binary Search

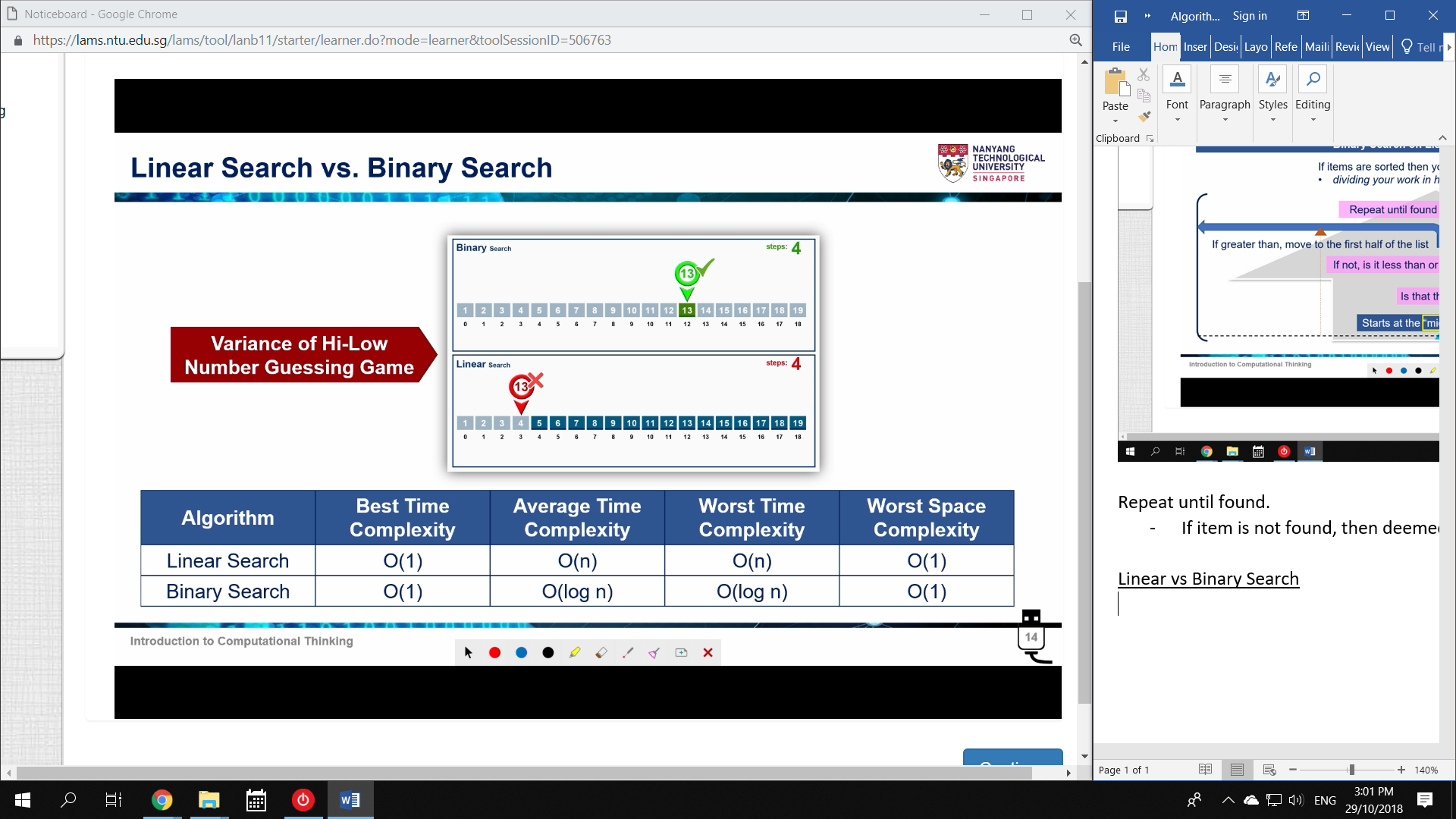
* Start at Middle of List.



Repeat until found.

* If item is not found, then deemed failure.

Linear vs Binary Search

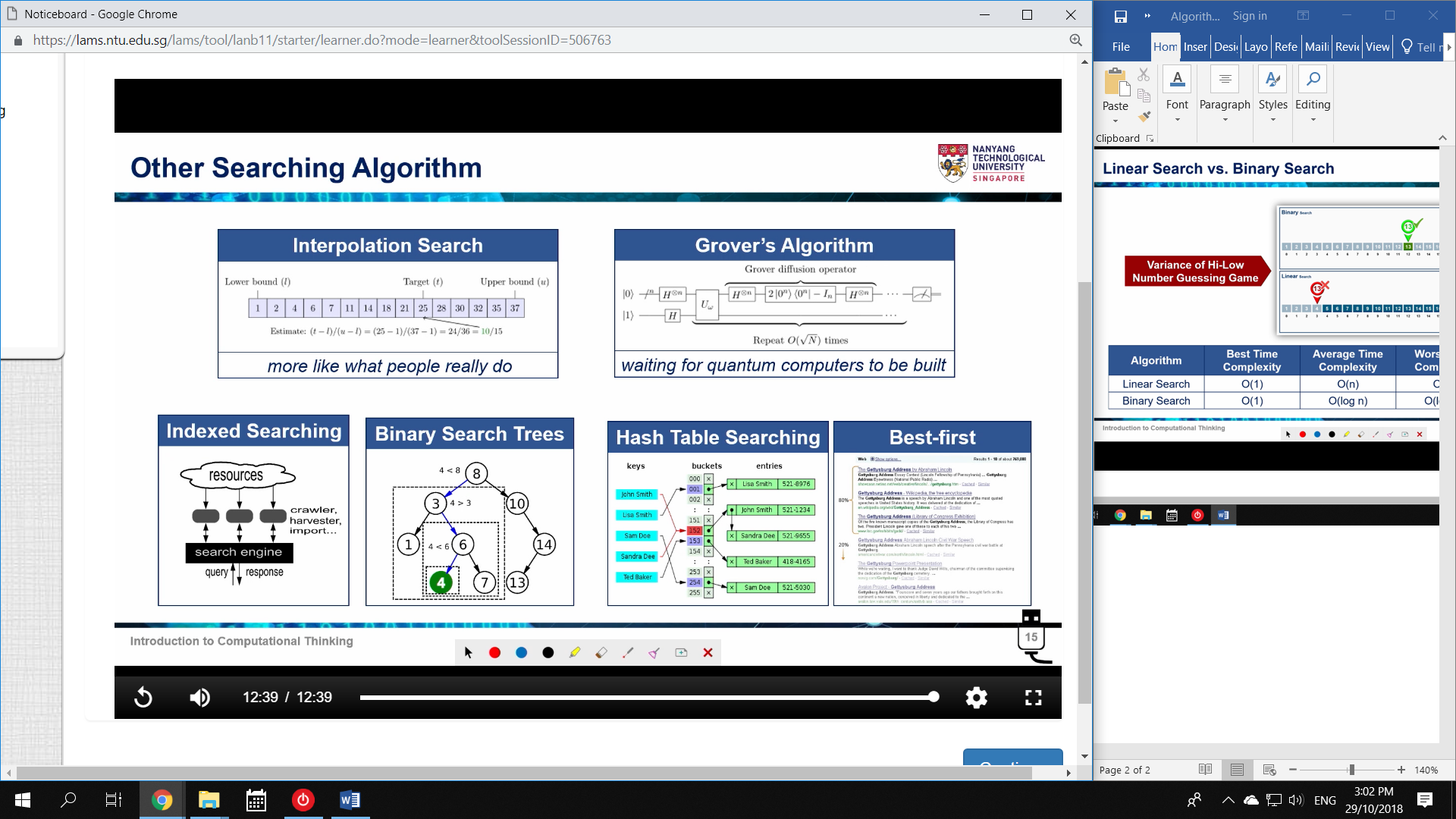


Worst Case (Linear): O(n)

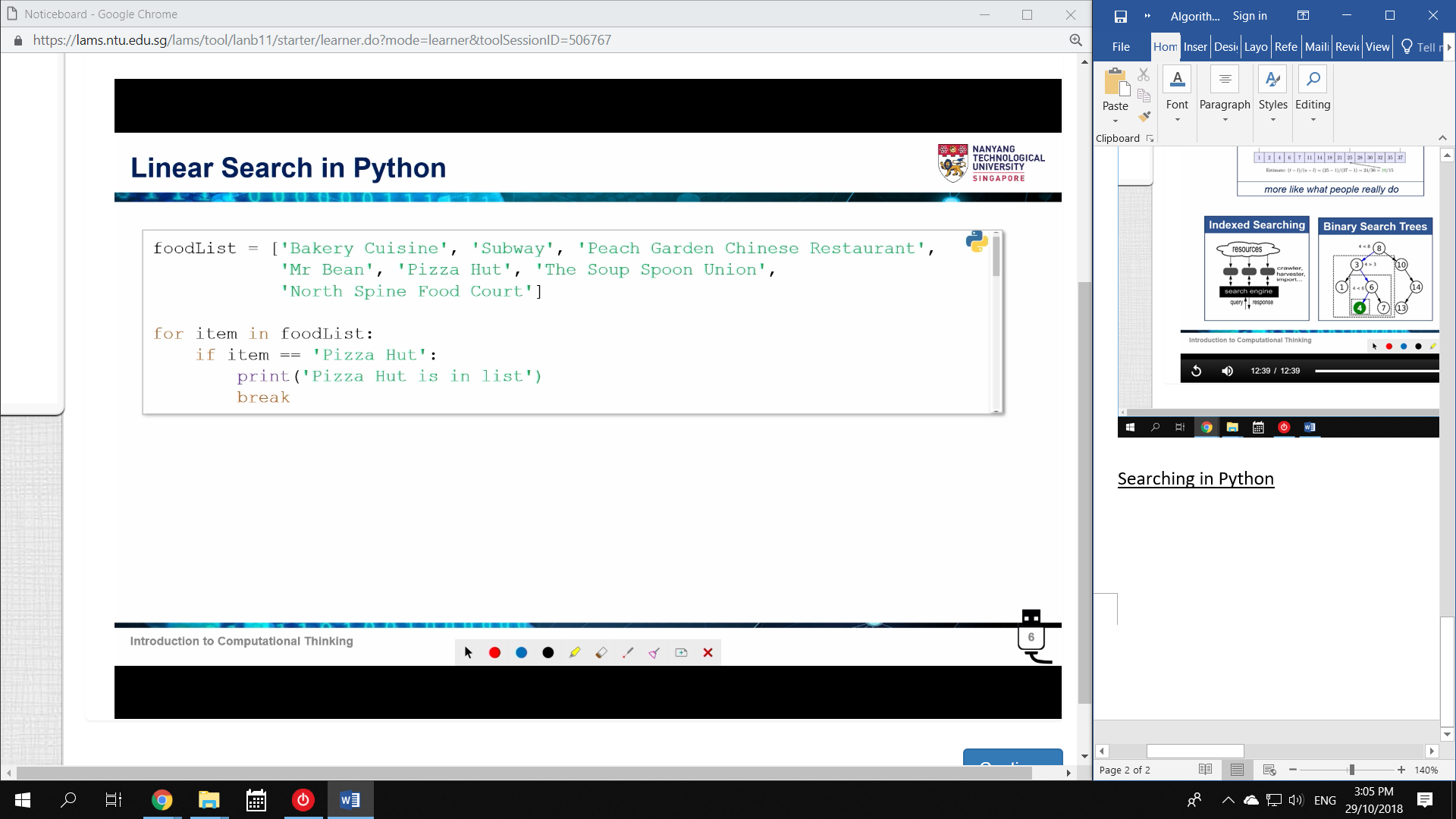
* ie., it takes n tries

Worst Case (Binary): O(log n)

* ie., it takes log(n) tries.



Searching in Python



Basic Iteration

*for item in foodList:*

*if item == ‘X’*

*print(‘X is present’)*

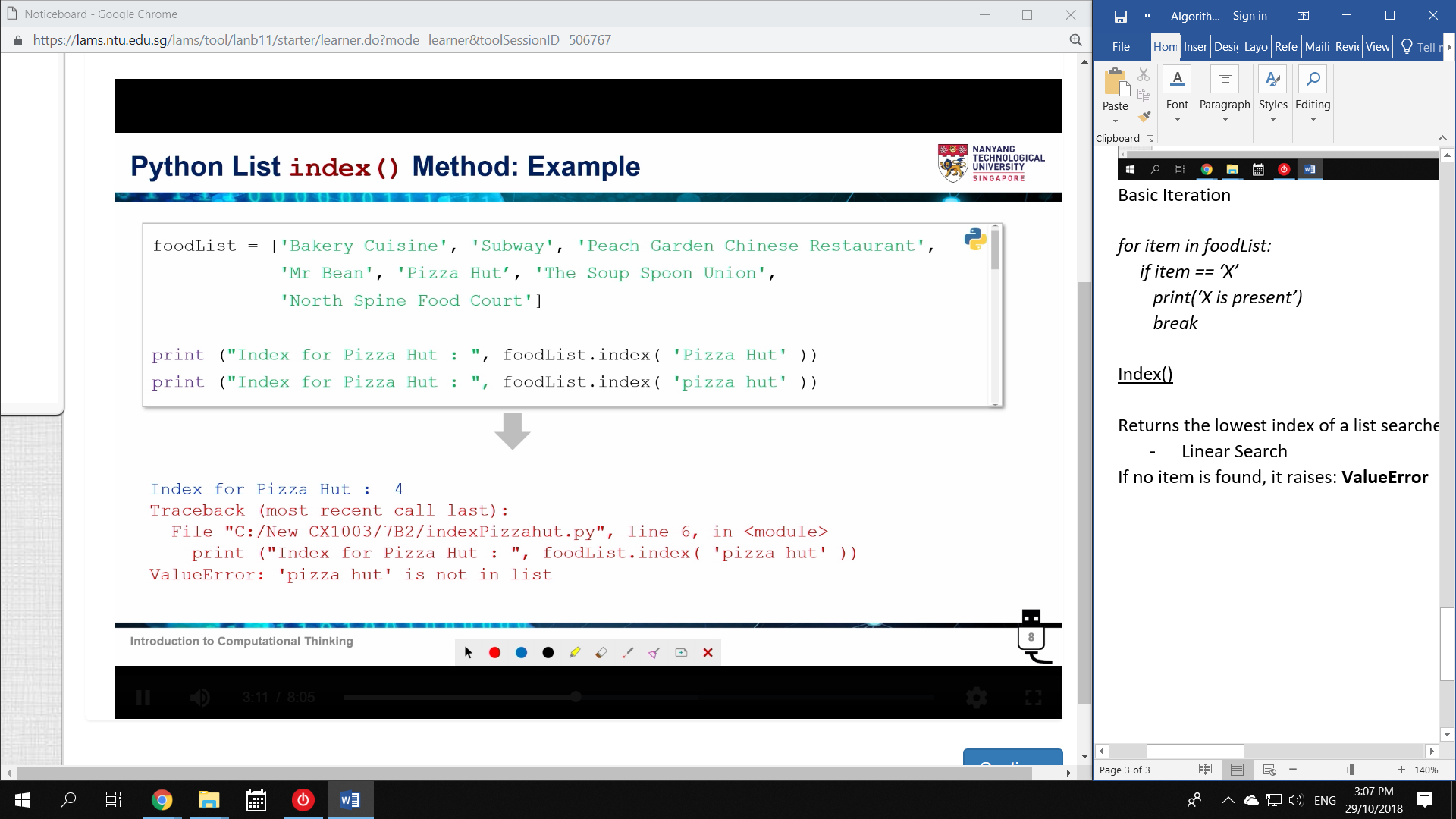
*break*

Index()

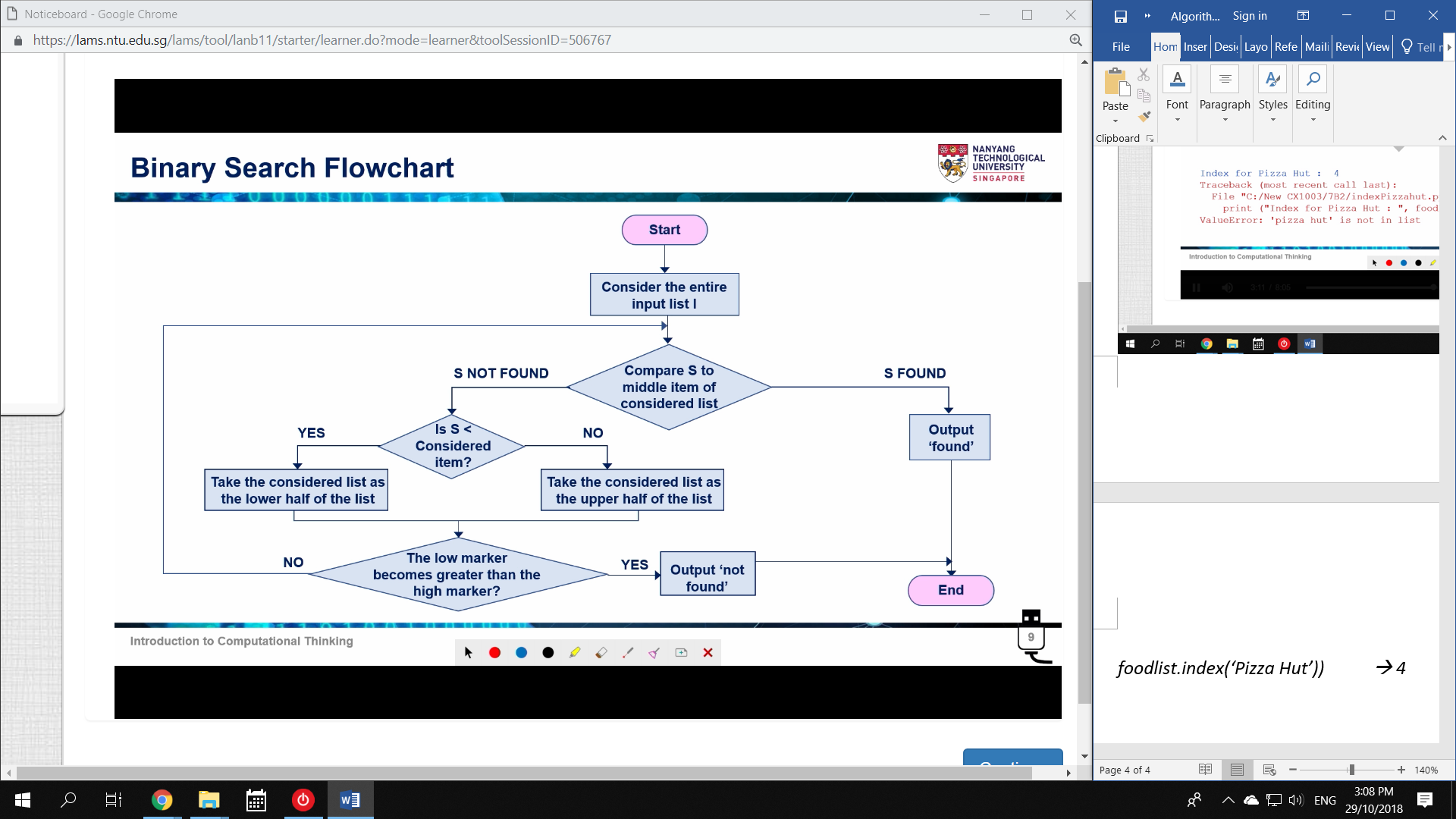
Returns the lowest index of a list searched.

* Linear Search

If no item is found, it raises: **ValueError**



*foodlist.index(‘Pizza Hut’)) 🡪 4*



1. Compare X to middle item **M**

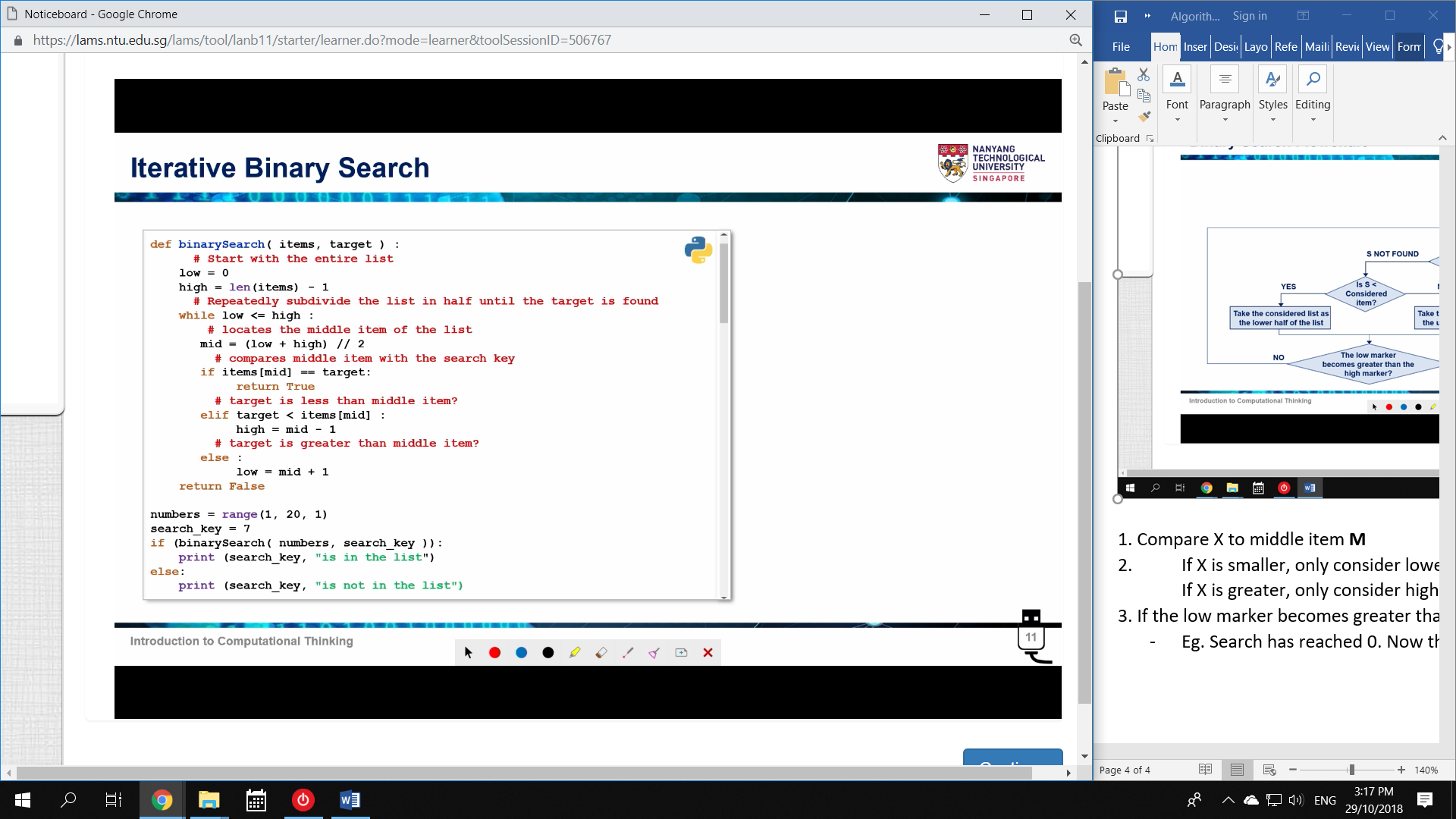
2. If X is smaller, only consider lower list (**0 to M-1**)

If X is greater, only consider higher list (**M+1 to n-1**) 🡪 n elements, [0] to [n-1]

3. If the low marker becomes greater than the high marker, Terminate

* Eg. Search has reached 0. Now the range is (0+1) to (0-1, or 1-1)

Iterative Code



Recursive Code

