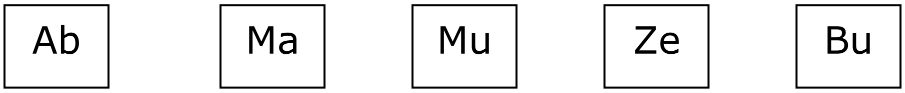
# Phonebook Design Challenge

*Collaborators*: Benjamin Eckley, Dave Ciolino-Volano, David Moste, Qianhui Vanessa Zou

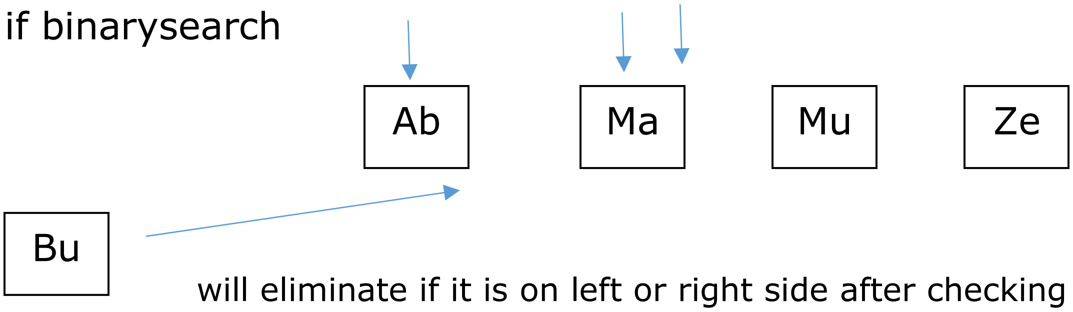
**Thought process:**

array list
linked list
-which takes longer from binary search?
array list may be faster, but depending on what happens under the hood
if same, linked list will be faster to add



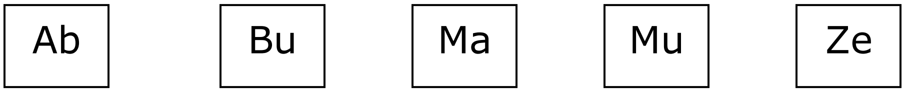
- do we do mergesort when adding at end or front?

if binarysearch

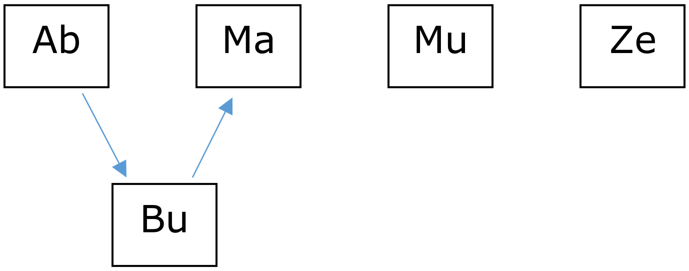


will eliminate if it is on left or right side after checking Ab as mid and high

to add: ArrayList will shift it down:



to add: LinkList will point without the shifting:



# Stucture:

The phonebook will contain two structures:

- An unsorted LinkedList
- A sometimes sorted ArrayList

## **add**:

The add method will simply add the new person to the front of a LinkedList and the end of an ArrayList. This has a constant runtime.

## **find**

We would use a mergeSort to sort the ArrayList and then a binary search to find the value. This will run in O(nLog(n)).

## **remove**

Remove can be done through any type of search and remove. We chose a linear search since the ArrayList may or may not be sorted at this point. This would need to be applied to both the LinkedList and the ArrayList. The runtime for remove would be O(n).

## **printList**

printList will run through the ArrayList and print every element. The runtime of printList would be O(n).