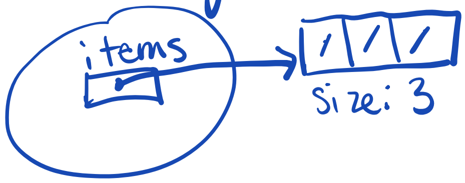
List ADT
Implementation

1. Choose data structure

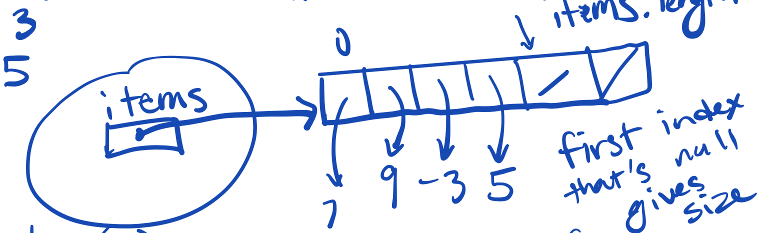
2. Implement using that d.s.

A. Array



isEmpty()
size()
insert()
remove(int index)
retrieve(int index)
add(Object item, int index)
void removeAll()

List
0 1 2 3
7 9 -3 5



if items.getIndex(0) = null → check if empty

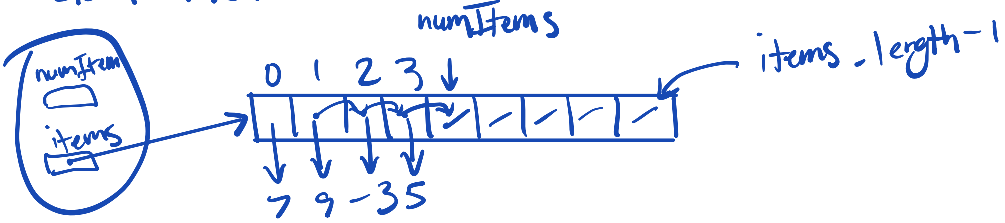
use while loop to count until null to get size

For add(), [0, size] params

remove() and retrieve(), [0, size-1] params

add datafield NumItems for size instead of looping through it. Everytime size changes, update datafield.

add



↑

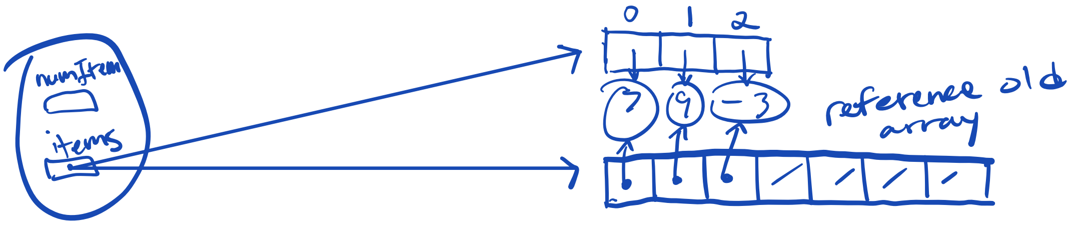
Shift all items to add new one

To remove, shift all back one and make index being removed = null

size of collection   size of array
numItems   ==   items.length

If array is full, can't add so check

If (full ...



Allocate memory for new array

replace max size checks with numItems == items.length

drop the null after loop in remove

ListArrayBasedPlus subclass of ListArrayBased

```
add
if full □
  { resize();
call the super
class add
  {
```

```
void resize ()
{
   Allocate memory for larger array
  copy from old array
 assign items the ref to old array
{
```

resize should add double old array

function that contains size of old array

inherits from object toString

```
void reverse()          String toString()
{                       {


}                       }
```

Iterate thru array with <u>for loop + DIA</u>

problem 2 - use ArrayList