# LinkedList Versus Array

## 1) Search

- Search operation yields the same result for both data structure
- ArrayList search operation is pretty fast compared to the LinkedList search operation
- We can use radom access with arrays: `getItem(int index)` which is O(1) time complexity
- LinkedList performance is O(N) time complexity
- So the concluin: ArrayList is better fot this operation
- Why?
- ArrayList maintains index based system fot its elements as it uses array data structure implicitly which makes it faster for searching an element in the list
- On the other hand LinkedList requires the traversal through all the items for searching an element

## 2) Deletion

- LinkedList remove operation takes O(1) time if we ewmove item from the beginning and usually this is the case
- ArrayList: removing first element (so at the beginning) takes O(N) time, removing the last item takes O(1) times
- But on average: we have to reconstruct the array when removing
- So the conclusion: LinkedList is better for this operation
- Why?

- LinkedList basically operates with pointers: removal only requires change in the pointer location whici can be done very fast

## 3) Memory management

- Arrays do not need any extra memory
- LinkedList on the other hand do need extra memory because of the references / pointer
- So in this aspect: arrays are better, they are memory friendly !!!

```
$ cd dillinger
$ npm install -d
$ node app
```

# Resume

| | Linked List | Arrays |
|---|---|---|
| Search | O(N) | O(1) |
| Insert at the start | O(1) | O(N) |
| Insert at the end | O(N) | O(1) |
| Waste space | O(N) | 0 |

Bruna Santos - January 27, 2018 10:55 am