

1 Data Structures

$$O(1) < O(\log(n)) < O(n^c) \text{ where } c < 1$$

$$O(n) < O(\log(n!)) = O(n \log(n)) < O(n^2)$$

$$O(n^k) [\text{ where } k > 2] < O(k^n) [\text{ where } k \geq 1] < O(n!)$$

1.1 Linked List

- Motivation: implementation of list using array needs to occupy contiguous memory space (can result in memory error)
- Variants of linked list:
 - Tailed (need to maintain head and tail)
 - Circular
 - Doubly linked (prev and next attributes for ListNode)
- How to find cycle?
Answer: use fast and slow pointers

```
1 slow = slow.next;
```

```
2 fast = fast.next.next;
```

- **[IMPT]** Drawing pictures is very important to visualize the program!

Java API: ArrayList or LinkedList

```
1 \\ constructor
2 ArrayList<Integer> list = new
  ArrayList<Integer>;
```

2 Algorithms

2.1 Sorting

3 Java Tricks

- Use StringBuilder for return statements
 - Java StringBuilder API
 - Zigzag conversion