Arrays

1. What is the difference between an array and a linked list?

- Array: Fixed size, random access.

- Linked List: Dynamic size, sequential access.

2. How do you find the largest and second largest element in an array?

Traverse the array once, keep track of `first` and `second` largest values.

3. What is the time complexity of inserting an element at the beginning, middle, and end of an array?

- Beginning: O(n)

- Middle: O(n)

- End: O(1) if space is available, else O(n) with reallocation.

Linked Lists

1. How do you reverse a linked list?

Iteratively update next pointers using three pointers: `prev`, `curr`, and `next`.

2. How do you detect a cycle in a linked list?

Use Floyd's Cycle Detection (Tortoise and Hare) algorithm.

3. How do you find the third node from the end?

Use two pointers:

- Move `first` pointer 3 steps ahead.

- Move both `first` and `second` one step until `first` is None.

- `second` is now the third from the end.

Python Code:

```
def third_from_end(head):
    first = head
    second = head
    for _ in range(3):
        if not first:
            return None
        first = first.next
    while first:
        first = first.next
        second = second.next
    return second
```