# COL106 Lab Week 2 Questions

**Problem 1: Finding cycle in Linked List**

Given the head element of a singly-linked list, return the node where the cycle begins (if it exists). If there is no cycle, return null.

A *cycle* is a sequence of nodes n1 n2 …. nk n1 such that n2 = next(n1), n3 = next(n2) and so on.

You can view the complete problem description here.

*Bonus: Can you try to implement this algorithm using O(1) space?*

**Problem 2: Next Greatest Element**

Given an array, print the *next greater element* for every element. Formally, for an element a[i], the next greater element is a[j] such that a[j] > a[i] and j is the smallest index such that j > i.

Elements for which no greater element exists, consider the next greater element as -1.

You can view the complete problem description here.

**Challenge Problem: Skip Lists**

A *skip list* is a probabilistic data structure which is based on sorted Linked Lists but provides O(logn) operations for search, insert, delete, etc. which are not possible in a regular linked list. (Why?)

You can read more about the problem here.

You will need to maintain multiple “levels” of linked lists, and randomly choose to promote elements to higher levels of the linked list to support these fast operations.

You can read about skip lists in general here.

You may choose to directly work on this problem if you think the others are *“easy”*. Those who complete it within the lab timings may get a special prize!

---