

Programmer's Study Group

Week 05

Last homework

- ▶ <https://leetcode.com/problems/remove-duplicates-from-sorted-list/>
 - ▶ <https://leetcode.com/problems/swap-nodes-in-pairs/>
 - ▶ <https://leetcode.com/problems/rotate-list/>
 - ▶ <https://leetcode.com/problems/partition-list/>
 - ▶ <https://leetcode.com/problems/reverse-linked-list-ii/>
 - ▶ <https://leetcode.com/problems/odd-even-linked-list/>
 - ▶ <https://leetcode.com/problems/intersection-of-two-linked-lists/>
 - ▶ <https://leetcode.com/problems/linked-list-cycle/>
 - ▶ <https://leetcode.com/problems/linked-list-cycle-ii/>
-
- ▶ <https://leetcode.com/problems/clone-graph/>
 - ▶ <https://leetcode.com/problems/course-schedule/>
 - ▶ <https://leetcode.com/problems/course-schedule-ii/>

Prove of <https://leetcode.com/problems/linked-list-cycle-ii/>

Using two pointers, one of them one step at a time. another pointer each take two steps. If there is a cycle, the slow and fast pointer will meet somewhere in the cycle.

Suppose the distance between the start node of list and the start node of cycle is s , the distance between start of cycle and the meet point is k , the length of the Cycle is r . When they meet, slow pointer walked n rounds of the circle, totally walked t steps from very beginning, faster pointer walked m rounds of the circle, totally walked $2t$ steps from very beginning.

$$\text{So } t = s + n * r + k, 2t = s + m * r + k \Rightarrow s = (m - 2n) * r - k$$

Since m and n are integer, we denote it as p .

$$\text{So } s = p * r - k$$

That means, using one pointer start from the start node of list, another pointer start from the first meet point, all of them walk one step at a time, the first time they meet each other is the start of the cycle.