**KMP Algorithm**

1. Haystack theke Needle vahir korar jonno Brute Force diye O(m\*n) ee nah kore O(m+n) ee kora possible ei algorithm diye
2. Link:

<https://youtu.be/0iDiUuHZE_g?si=_nyJqFFH7_fm4Pj7>

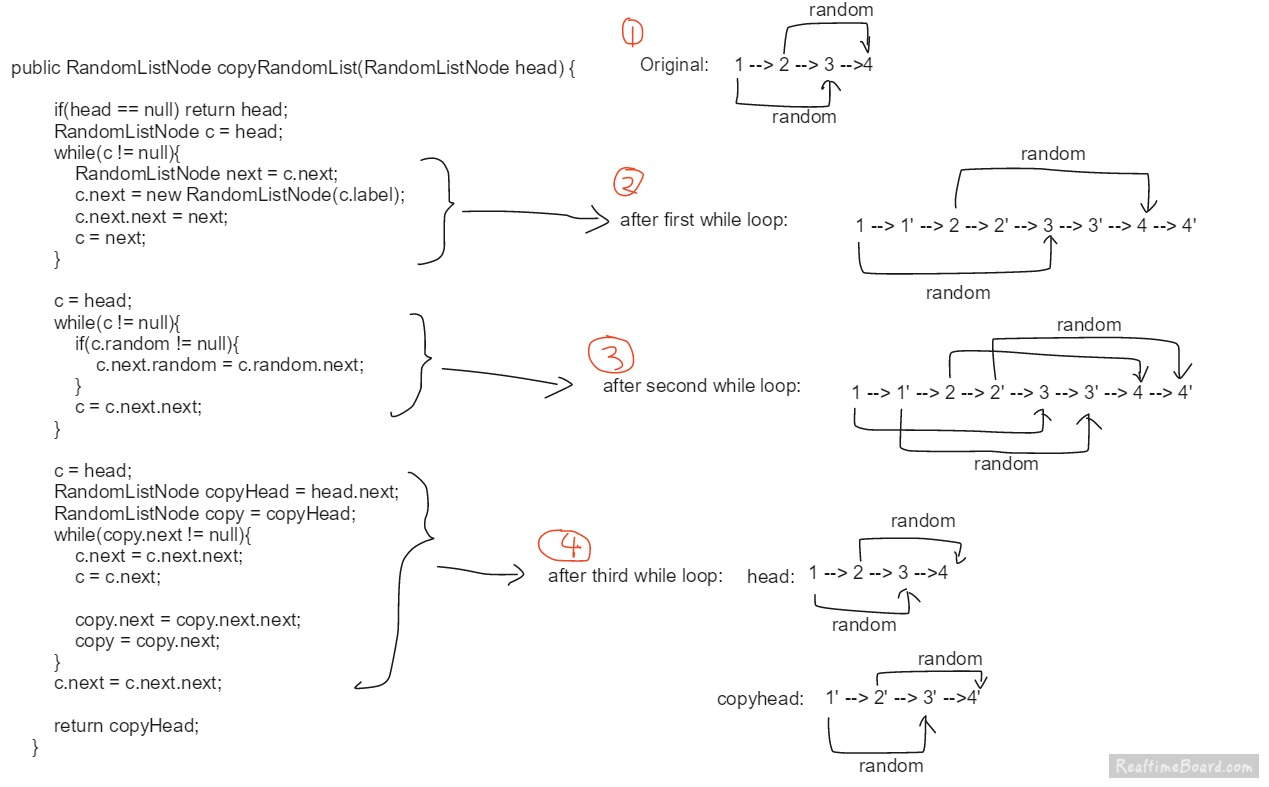
**Linked List**

1. always → **ListNode\* DummyNode=new ListNode(0, head);** eirokom ekta DummyNode use korbo…ete kore Linked List er “Starting/Initial” Position er jonno kono Condition alada kore “**nullptr or NULL**” check kora lagbe nah… amra just “DummyNode->next” return kore dibo **NewHead** hisave
2. **nullptr** is a keyword that can be used at all places where **NULL** is expected. Like NULL, nullptr is implicitly convertible and comparable to any pointer type. Unlike NULL, it is not implicitly convertible or comparable to integral types.
3. Link:

<https://youtu.be/RF_M9tX4Eag?si=4URB5jV6rTKoe90Y>

**Problem:**

<https://leetcode.com/problems/copy-list-with-random-pointer/description/>



**Vector Delete a Middle Element in O(1) time**

If a element is in index “k”... then just **SWAP** the “kth” index Element with the **LAST** element… then just do “**v.pop\_back()**”

**Kadane's Algorithm**

**Maximum Subarray** Calculation Algorithm

**Array**

**3 types** of Questions for Arrays:

1. Two Pointer
2. Sliding Window → “Subarray” ei Word ta shunlei “Sliding Window” marar try korbo… In Sliding Window either the “Window” will “Shrink” or “Stretch”... In this Problem, we **ITERATE** the “**Right Pointer**” first
3. Prefix Sum

## **Sliding Window**

In Sliding Window either the “Window” will “Shrink” or “Stretch”... In this Problem, we **ITERATE** the “**Right Pointer**” first and move the Left Pointer with Left pointer++

int left=0;

for(int right=0;right<nums.size();right++){

…….

…….

}

**Hare & Tortoise**

Slow Pointer & Fast Pointer nibo to calculate if a Graph has CYCLE or Not

Probable Mistake while coding:

fast = **fast**->next->next**; ❌**

fast = **current**->next->next; ✅

**Wrong Code**

while(true){

slow = **current**;

fast = **current**->next->next;

}

This is a Mistake… chinta kore dekh… ei Code er jonno Fast Pointer always Slow Pointer theke 2 ghor agai takbe… code hobe ashole… **Fast Pointer** er “**current Faster Pointer” er Position theke amra 2 ghor samne jabo**

**Correct Code**

slow = current;

fast = current;

while(true){

slow = **slow**->next;

fast = **fast**->next->next;

}