In [ ]:
```
https://leetcode.com/problems/missing-number/
https://leetcode.com/problems/single-number/
https://leetcode.com/problems/decode-xored-array/
https://leetcode.com/problems/prison-cells-after-n-days/
```

In [ ]:

Question
https://leetcode.com/problems/merge-two-sorted-lists/ (https://leetcode.com/problems/merge-two-sorted-lists/)

In [ ]:

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {

        ListNode* res;
        ListNode* curr;

        if (list1 == NULL) {
            return list2;
        }

        if (list2 == NULL) {
            return list1;
        }

        // Both lists have atleast 1 value
        if (list1->val < list2->val) {
            res = list1;
            curr = list1;
            list1=list1->next;
        } else {
            res = list2;
            curr = list2;
            list2=list2->next;
        }

        while(list1 != NULL && list2 != NULL) {
            if (list1->val < list2->val) {
                curr->next = list1;
                curr = list1;
                list1=list1->next;
            } else {
                curr->next = list2;
                curr = list2;
                list2=list2->next;
            }
        }

        if (list1 != NULL) {
            curr ->next =  list1;
        }
        if (list2 != NULL) {
            curr ->next =  list2;
        }

        return res;
    }
```

```
};
```

In [ ]:

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {

        if (list1 == NULL) {
            return list2;
        }

        if (list2 == NULL) {
            return list1;
        }

        ListNode*curr = NULL;
        // Both lists have atleast 1 value
        if (list1->val < list2->val) {
            curr = list1;
            list1=list1->next;
        } else {
            curr = list2;
            list2=list2->next;
        }

        curr->next = mergeTwoLists(list1, list2);
        return curr;
    }
};
```

In [ ]:

Question

https://leetcode.com/problems/missing-number/ (https://leetcode.com/problems/missing-number/)

In [ ]:
```java
class Solution {
    public int missingNumber(int[] nums) {
        int n = nums.length;

        int sum = Arrays.stream(nums).sum();

        return ((n * (n+1))/2) - sum;
    }
}
```

In [ ]:
```java
class Solution {
    public int missingNumber(int[] nums) {

        int n= nums.length;
        int expectedTotal = (n*(n+1))/2;

        int total=0;
        for(int num:nums){
            total += num;
        }

        return expectedTotal-total;

    }
}
```

In [ ]:
```java
class Solution {
    public int missingNumber(int[] nums) {
        int n=nums.length;
        System.out.println(n);
        int total_sum = (n*(n+1))/2;
        System.out.println(total_sum);
        int array_sum = 0;
        for(int i=0; i<n;i++){
            array_sum+=nums[i];
        }
        int missing = total_sum-array_sum;
        return missing;
    }
}
```

In [ ]:
```python
class Solution:
    def missingNumber(self, nums: List[int]) -> int:
        n = sum(nums)
        m = sum(range(0,len(nums)+1))
        return m-n
```

In [ ]:
```java
public class Solution {
    public int missingNumber(int[] nums) {

        int n = nums.length;
        int sumOfAll = n * (n + 1) / 2;

        for (int i : nums) {
            sumOfAll -= i;
        }

        return sumOfAll;
    }
}
```

In [ ]:
```java
class Solution {
    public int missingNumber(int[] nums) {

        int b= nums.length;
        int expectedTotal = (b*(b+1))/2;

        int total=0;
        for(int num:nums){
            total += num;
        }

        return expectedTotal-total;

    }
}
```

In [ ]:
```java
class Solution {
 public int missingNumber(int[] nums) {
 int count[] = new int[nums.length+1];
 for(int i=0;i<nums.length;i++){
 count[nums[i]]++;
 }
 for(int i=0;i<=nums.length;i++){
 if(count[i]==0) return i;
 }
 return -1;
 }
}
```

In [ ]:
```java
class Solution {
    public int missingNumber(int[] nums) {

        int n= nums.length;
        int expectedTotal = (n*(n+1))/2;

        int total=0;
        for(int num:nums){
            total += num;
        }

        return expectedTotal-total;

    }
}
```

In [ ]:
```
```

Question

https://leetcode.com/problems/single-number/ (https://leetcode.com/problems/single-number/)

In [ ]:
```python
class Solution:
    def singleNumber(self, nums: List[int]) -> int:
        nums.sort() # O(n log n)
        for i in range(0, len(nums) - 1, 2):
            if nums[i] != nums[i+1]:
                return nums[i]

        return nums[-1]
```

In [ ]:
```java
class Solution {
public:
    int singleNumber(vector<int>& nums) {
        set<int> s;

        for (auto num: nums) {
            if (s.count(num) == 0) {
                s.insert(num);
            } else {
                s.erase(num);
            }
        }

        return *s.begin();
    }
};
```

In [ ]:
```java
class Solution {
    public int singleNumber(int[] nums) {

        int xor = 0;

        for( int num:nums){
            xor = xor^num;
        }

        return xor;
    }
}
```

In [ ]:
```python
class Solution:
    def singleNumber(self, nums: List[int]) -> int:
        xor = 0
        for n in nums:
            xor = xor ^ n
        return xor
```

In [ ]:
```java
class Solution {
    public int singleNumber(int[] nums) {
        int single=0;
        for(int i=0; i<nums.length;i++){
            single=single^nums[i];
        }
        return single;
    }
}
```

In [ ]:
```
```

Question

https://leetcode.com/problems/decode-xored-array/ (https://leetcode.com/problems/decode-xored-array/)

In [ ]:
```java
class Solution {
    public int[] decode(int[] encoded, int first) {
        int[] arr = new int[encoded.length+1];
        arr[0] = first;
        for (int i = 1; i<encoded.length+1;i++){
            arr[i] = arr[i-1] ^ encoded[i-1];
        }
        return arr;
    }
}
```

In [ ]:
```python
class Solution:
    def decode(self, encoded: List[int], first: int) -> List[int]:
        res = [first]
        for i in encoded:
            res.append(res[-1]^i)
        return res
```

In [ ]:
```python

```

Question

https://leetcode.com/problems/prison-cells-after-n-days/ (https://leetcode.com/problems/prison-cells-after-n-days/)