Assignment Solutions | Cyclic Sort | Week 13

1. What is the worst case time complexity of cycle sort?

   a) O(n)

   b) O(log n)

   c) O(n log n)

   d) O(n*n)

Solution:

`a) O(n)`

2. You have a set of integers `s`, which originally contains all the numbers from `1` to `n`. Unfortunately, due to some error, one of the numbers in `s` got duplicated to another number in the set, which results in **repetition of one** number and **loss of another** number.

   You are given an integer array `nums` representing the data status of this set after the error.

   Find the number that occurs twice and the number that is missing and return them in the form of an array.   [Leetcode 645]

   Example 1:

   Input: nums = [1,2,2,4]

   Output: [2,3]

   Example 2:

   Input: nums = [1,1]

   Output: [1,2]

Solution:

```
class Solution {
public:
    vector<int> findErrorNums(vector<int>& a) {
        int i = 0;
        int n = a.size();
        for(int i=0;i<n;i++){
            while(a[i] != a[a[i]-1])swap(a[i] , a[a[i]-1]);
        }

        for(int i=0;i<a.size();i++){
            if(i+1 != a[i]){
                return {a[i] , i + 1};
            }
        }
        return {};
    }
};
```

3. Given an integer array `nums` of length `n` where all the integers of `nums` are in the range `[1, n]` and each integer appears **once** or **twice**, return *an array of all the integers that appears **twice***.

    You must write an algorithm that runs in `O(n)` time and uses only constant extra space.

    Example 1:

    Input: nums = [4,3,2,7,8,2,3,1]

    Output: [2,3]

    Example 2:

    Input: nums = [1,1,2]

    Output: [1]

    Example 3:

    Input: nums = [1]

    Output: []

Solution:

```cpp
class Solution {
public:
    vector<int> findDuplicates(vector<int>& a) {
        int n = a.size();
        for(int i=0;i<n;i++){
            while(a[i] != a[a[i]-1])swap(a[i] , a[a[i]-1]);
        }
        vector<int>b;
        for(int i=0;i<n;i++){
            if(i+1 != a[i])b.push_back(a[i]);
        }
        return b;
    }
};
```