DSA UPDATES

8.1.24

1) <https://leetcode.com/problems/longest-common-prefix/description/>

class Solution {

    public String longestCommonPrefix(String[] strs) {

        Arrays.sort(strs);

        String a = strs[0];

        String b = strs[strs.length-1];

        int i =0;

        while(i<a.length()&& i<b.length()){

            if(a.charAt(i)==b.charAt(i)){

                i++;

            }

            else{

                break;

            }

        }

        return a.substring(0,i);

    }

}

2) <https://leetcode.com/problems/rotate-array/>

class Solution {

    public void rotate(int[] nums, int k) {

        k = k%nums.length;

        reverse(nums,0, nums.length-1);

        reverse(nums,0, k-1);

        reverse(nums,k,nums.length-1);

    }

    public void reverse(int[] nums, int st, int end){

        while(st<end){

            int temp = nums[st];

            nums[st] = nums[end];

            nums[end] = temp;

            st++;

            end--;

        }

    }

}

02.08.24

1) <https://leetcode.com/problems/minimum-swaps-to-group-all-1s-together-ii/description/?envType=daily-question&envId=2024-08-02>

class Solution {

    public int minSwaps(int[] nums) {

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

        int n = nums.length;

        int cnt = 0;

        for (int i = 0; i < k; ++i) {

            cnt += nums[i];

        }

        int mx = cnt;

        for (int i = k; i < n + k; ++i) {

            cnt += nums[i % n] - nums[(i - k + n) % n];

            mx = Math.max(mx, cnt);

        }

        return k - mx;

    }

}

2) <https://leetcode.com/problems/max-consecutive-ones/description/>

class Solution {

    public int findMaxConsecutiveOnes(int[] nums) {

        int count =0;

        int ans = 0;

        for(int i =0;i<nums.length;i++){

            if(nums[i]==1){

                count++;

                 ans = Math.max(ans,count);

            }

            else{

                count=0;

            }

        }return ans;

    }

}