[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class NearlySimilarRectangles {

public static long nearlySimilarRectangles(List<List<Long>> sides) {

Map<String, Integer> map = new HashMap<>();

for (List<Long> side : sides) {

long w = side.get(0);

long h = side.get(1);

long gcd = gcd(w, h);

String key = (w / gcd) + "," + (h / gcd);

map.put(key, map.getOrDefault(key, 0) + 1);

}

long count = 0;

for (int value : map.values()) {

count += (long) value \* (value - 1) / 2;

}

return count;

}

public static long gcd(long a, long b) {

return b == 0 ? a : gcd(b, a % b);

}

}

import java.util.\*;

public class NearlySimilarRectangles {

public static long nearlySimilarRectangles(List<List<Long>> sides) {

Map<String, Integer> map = new HashMap<>();

for (List<Long> side : sides) {

long w = side.get(0);

long h = side.get(1);

long gcd = gcd(w, h);

String key = (w / gcd) + "," + (h / gcd);

map.put(key, map.getOrDefault(key, 0) + 1);

}

long count = 0;

for (int value : map.values()) {

count += (long) value \* (value - 1) / 2;

}

return count;

}

public static long gcd(long a, long b) {

return b == 0 ? a : gcd(b, a % b);

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class VowelSubstring {

public static String findVowelSubstring(String s, int k) {

Set<Character> vowels = new HashSet<>(Arrays.asList('a', 'e', 'i', 'o', 'u'));

int cur = 0, best = 0, ans = 0;

for (int i = 0; i < k; i++) {

if (vowels.contains(s.charAt(i))) {

cur++;

}

}

best = cur;

for (int i = k; i < s.length(); i++) {

if (vowels.contains(s.charAt(i))) {

cur++;

}

if (vowels.contains(s.charAt(i - k))) {

cur--;

}

if (cur > best) {

best = cur;

ans = i - k + 1;

}

}

if (best > 0) {

return s.substring(ans, ans + k);

} else {

return "Not found!";

}

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class MostActiveCustomers {

public static List<String> mostActive(List<String> customers) {

Map<String, Integer> map = new HashMap<>();

for (String c : customers) {

map.put(c, map.getOrDefault(c, 0) + 1);

}

List<String> result = new ArrayList<>();

int totalCustomers = customers.size();

for (Map.Entry<String, Integer> entry : map.entrySet()) {

if ((double) entry.getValue() / totalCustomers >= 0.05) {

result.add(entry.getKey());

}

}

Collections.sort(result);

return result;

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class MostBalancedPartition {

public static int mostBalancedPartition(int[] parent, int[] filesSize) {

int n = parent.length;

List<List<Integer>> children = new ArrayList<>();

for (int i = 0; i < n; i++) {

children.add(new ArrayList<>());

}

for (int i = 1; i < n; i++) {

children.get(parent[i]).add(i);

}

int[] sizeSums = new int[n];

sizeSumsRec(0, children, filesSize, sizeSums);

int totalSize = sizeSums[0];

int minDifference = Integer.MAX\_VALUE;

for (int i = 1; i < n; i++) {

minDifference = Math.min(minDifference, Math.abs(totalSize - 2 \* sizeSums[i]));

}

return minDifference;

}

private static int sizeSumsRec(int i, List<List<Integer>> children, int[] filesSize, int[] sizeSums) {

sizeSums[i] = filesSize[i];

for (int child : children.get(i)) {

sizeSums[i] += sizeSumsRec(child, children, filesSize, sizeSums);

}

return sizeSums[i];

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class PossibleChanges {

public static List<String> possibleChanges(List<String> usernames) {

List<String> ans = new ArrayList<>();

for (String u : usernames) {

if (u.length() <= 1) {

ans.add("NO");

continue;

}

boolean found = false;

for (int i = 0; i < u.length() - 1; i++) {

if (u.charAt(i) > u.charAt(i + 1)) {

ans.add("YES");

found = true;

break;

}

}

if (!found) {

ans.add("NO");

}

}

return ans;

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class FilledOrders {

public static int filledOrders(List<Integer> order, int k) {

Collections.sort(order);

int ans = 0;

for (int x : order) {

if (x <= k) {

ans++;

k -= x;

} else {

break;

}

}

return ans;

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class FindSum {

public static List<Integer> findSum(List<Integer> numbers, List<int[]> queries) {

int n = numbers.size();

int[] a = new int[n + 1];

int[] b = new int[n + 1];

for (int i = 0; i < n; i++) {

a[i + 1] = a[i] + numbers.get(i);

b[i + 1] = b[i] + (numbers.get(i) == 0 ? 1 : 0);

}

List<Integer> result = new ArrayList<>();

for (int[] query : queries) {

int l = query[0];

int r = query[1];

int x = query[2];

result.add(a[r] - a[l - 1] + x \* (b[r] - b[l - 1]));

}

return result;

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class StringAnagram {

public static List<Integer> stringAnagram(List<String> dictionary, List<String> query) {

Map<String, Integer> map = new HashMap<>();

for (String w : dictionary) {

String sortedWord = sortString(w);

map.put(sortedWord, map.getOrDefault(sortedWord, 0) + 1);

}

List<Integer> ans = new ArrayList<>();

for (String w : query) {

String sortedWord = sortString(w);

ans.add(map.getOrDefault(sortedWord, 0));

}

return ans;

}

private static String sortString(String s) {

char[] chars = s.toCharArray();

Arrays.sort(chars);

return new String(chars);

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class DecryptPassword {

public static String decryptPassword(String s) {

char[] chars = s.toCharArray();

int i = 0;

while (i < chars.length && Character.isDigit(chars[i]) && chars[i] != '0') {

i++;

}

List<Integer> zeroPositions = new ArrayList<>();

for (int j = i; j < chars.length; j++) {

if (chars[j] == '0') {

zeroPositions.add(j);

}

}

for (int j = 0; j < zeroPositions.size(); j++) {

chars[zeroPositions.get(j)] = chars[i - j - 1];

}

for (int j = i; j < chars.length; j++) {

if (chars[j] == '\*') {

char temp = chars[j - 1];

chars[j - 1] = chars[j - 2];

chars[j - 2] = temp;

}

}

return new String(chars, i, chars.length - i).replace("\*", "");

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class MaxCost {

public static int maxCost(List<Integer> cost, List<String> labels, int dailyCount) {

int ans = 0;

int curCnt = 0;

int curCost = 0;

for (int i = 0; i < cost.size(); i++) {

curCost += cost.get(i);

if (labels.get(i).equals("illegal")) {

continue;

}

curCnt++;

if (curCnt == dailyCount) {

ans = Math.max(ans, curCost);

curCnt = 0;

curCost = 0;

}

}

return ans;

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class MinTime {

public static int minTime(List<Integer> files, int numCores, int limit) {

List<Integer> x = new ArrayList<>();

List<Integer> y = new ArrayList<>();

for (int f : files) {

if (f % numCores == 0) {

x.add(f);

} else {

y.add(f);

}

}

Collections.sort(x, Collections.reverseOrder());

int totalTime = 0;

for (int i = 0; i < limit && i < x.size(); i++) {

totalTime += x.get(i) / numCores;

}

for (int i = limit; i < x.size(); i++) {

totalTime += x.get(i);

}

for (int f : y) {

totalTime += f;

}

return totalTime;

}

}

[13/01, 12:21] Akshitha Batch1 Capgemini Cr: import java.util.\*;

public class LongestSubarray {

public static int longestSubarray(int[] arr) {

int n = arr.length;

int ans = 0;

for (int i = 0; i < n; i++) {

List<Integer> w = new ArrayList<>();

int cnt = 0;

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

if (w.contains(arr[j])) {

cnt++;

continue;

}

if (w.isEmpty()) {

w.add(arr[j]);

} else if (w.size() == 1) {

if (Math.abs(w.get(0) - arr[j]) > 1) {

break;

} else {

w.add(arr[j]);

}

} else {

break;

}

cnt++;

}

ans = Math.max(ans, cnt);

}

return ans;

}

}