

Maximum subset XOR

Hard Accuracy: **28.93%** Submissions: **21K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `arr[]` of N positive integers. Find an integer denoting the **maximum XOR subset** value in the given array `arr[]`.

Example 1:

Input :

$N = 3$

`arr[] = {2, 4, 5}`

Output : 7

Explanation :

The subset `{2, 5}` has maximum subset XOR value.

Example 2 :

Input :

$N = 3$

`arr[] = {9, 8, 5}`

Output : 13

Explanation :

The subset `{8, 5}` has maximum subset XOR value.

Your Task :

You don't need to read input or print anything. Your task is to complete the function **maxSubsetXOR()** which takes the array and an integer as input and returns the maximum subset XOR value.

Expected Time Complexity : $O(N \cdot \log(\max(\text{arr}[i])))$

Expected Auxiliary Space : $O(1)$

Constraints :

$1 \leq N \leq 10^5$

$1 \leq \text{arr}[i] \leq 10^6$

Smallest range in K lists

Hard Accuracy: **43.21%** Submissions: **18K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given **K** sorted lists of integers, **KSortedArray[]** of size **N** each. The task is to find the smallest range that includes at least one element from each of the **K** lists. If more than one such range's are found, return the first such range found.

Example 1:

Input:

$N = 5, K = 3$

$\text{KSortedArray}[][] = \{\{1\ 3\ 5\ 7\ 9\},$
 $\{0\ 2\ 4\ 6\ 8\},$
 $\{2\ 3\ 5\ 7\ 11\}\}$

Output: 1 2

Explanation: $K = 3$

A:[1 3 5 7 9]

B:[0 2 4 6 8]

C:[2 3 5 7 11]

Smallest range is formed by number 1
present in first list and 2 is present
in both 2nd and 3rd list.

Example 2:

Input:

N = 4, K = 3

KSortedArray[][] = {{1 2 3 4},
 {5 6 7 8},
 {9 10 11 12}}

Output: 4 9

Your Task :

Complete the function *findSmallestRange()* that receives array , array size n and k as parameters and returns the output range (as a pair in cpp and array of size 2 in java and python)

Expected Time Complexity : $O(n * k * \log k)$

Expected Auxilliary Space : $O(k)$

Constraints:

$1 \leq K, N \leq 500$

$0 \leq a[i] \leq 10^5$

Max Circular Subarray Sum

Hard Accuracy: **29.37%** Submissions: **77K** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `arr[]` of `N` integers arranged in a **circular** fashion. Your task is to find the **maximum contiguous subarray sum**.

Example 1:

Input:

`N = 7`

`arr[] = {8,-8,9,-9,10,-11,12}`

Output:

22

Explanation:

Starting from the last element of the array, i.e, 12, and moving in a circular fashion, we have max subarray as 12, 8, -8, 9, -9, 10, which gives maximum sum as 22.

Example 2:

Input:

`N = 8`

`arr[] = {10,-3,-4,7,6,5,-4,-1}`

Output:

23

Explanation: Sum of the circular subarray with maximum sum is 23

Your Task:

The task is to complete the function `circularSubarraySum()` which returns a sum of the circular subarray with maximum sum.

Expected Time Complexity: $O(N)$.

Expected Auxiliary Space: $O(1)$.

Constraints:

$1 \leq N \leq 10^6$

$-10^6 \leq \text{Arr}[i] \leq 10^6$

Max Circular Subarray Sum

Hard Accuracy: **29.37%** Submissions: **77K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `arr[]` of N integers arranged in a **circular** fashion. Your task is to find the **maximum contiguous subarray sum**.

Example 1:

Input:

$N = 7$

`arr[] = {8,-8,9,-9,10,-11,12}`

Output:

22

Explanation:

Starting from the last element of the array, i.e, 12, and moving in a circular fashion, we have max subarray as 12, 8, -8, 9, -9, 10, which gives maximum sum

as 22.

Example 2:**Input:**

N = 8

arr[] = {10,-3,-4,7,6,5,-4,-1}

Output:

23

Explanation: Sum of the circular subarray with maximum sum is 23

Your Task:

The task is to complete the function `circularSubarraySum()` which returns a sum of the circular subarray with maximum sum.

Expected Time Complexity: $O(N)$.

Expected Auxiliary Space: $O(1)$.

Constraints:

$1 \leq N \leq 10^6$

$-10^6 \leq \text{Arr}[i] \leq 10^6$

Next Smallest Palindrome

Hard Accuracy: **19.63%** Submissions: **16K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a number, in the form of an array **Num[]** of size **N** containing digits from **1 to 9**(inclusive). The task is to find the **next smallest palindrome strictly larger than the given number**.

Example 1:**Input:**

N = 11

Num[] = {9, 4, 1, 8, 7, 9, 7, 8, 3, 2, 2}

Output: 9 4 1 8 8 0 8 8 1 4 9**Explanation:** Next smallest palindrome is 94188088149.**Example 2:****Input:**

N = 5

Num[] = {2, 3, 5, 4, 5}

Output: 2 3 6 3 2**Explanation:** Next smallest palindrome is 23632.**Your Task:**

Complete the function **generateNextPalindrome()** which takes an array **num**, and an single integer **n**, as input parameters and returns an array of integers denoting the answer. You don't to print answer or take inputs.

Expected Time Complexity: $O(N)$ **Expected Auxiliary Space:** $O(1)$ **Constraints:** $1 \leq N \leq 10^5$ $1 \leq \text{Num}[i] \leq 9$

Count Smaller elements

Hard Accuracy: **38.16%** Submissions: **34K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array **Arr** of size **N** containing positive integers. Count number of smaller elements on right side of each array element.

Example 1:

Input:

N = 7

Arr[] = {12, 1, 2, 3, 0, 11, 4}

Output: 6 1 1 1 0 1 0

Explanation: There are 6 elements right after 12. There are 1 element right after 1. And so on.

Example 2:

Input:

N = 5

Arr[] = {1, 2, 3, 4, 5}

Output: 0 0 0 0 0

Explanation: There are 0 elements right after 1. There are 0 elements right after 2. And so on.

Your Task:

You don't need to read input or print anything. Your task is to complete the

function **constructLowerArray()** which takes the array of integers **arr** and **n** as parameters and returns an array of integers denoting the answer.

Expected Time Complexity: $O(N \cdot \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 10^6$

$0 \leq \text{Arr}_i \leq 10^8$

Largest rectangle of 1s with swapping of columns allowed

Hard Accuracy: **55.1%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a matrix **mat** of size **R*C** with 0 and 1s, find the largest rectangle of all 1s in the matrix. The rectangle can be formed by swapping any pair of columns of given matrix.

Example 1:

Input:

R = 3, C = 5

mat[][] = {{0, 1, 0, 1, 0},
 {0, 1, 0, 1, 1},
 {1, 1, 0, 1, 0}};

Output: 6

Explanation: The largest rectangle's area is 6. The rectangle can be formed by swapping column 2 with 3. The matrix after swapping will be

0 0 1 1 0

```
0 0 1 1 1
1 0 1 1 0
```

Example 2:**Input:**

R = 4, C = 5

```
mat[][] = {{0, 1, 0, 1, 0},
            {0, 1, 1, 1, 1},
            {1, 1, 1, 0, 1},
            {1, 1, 1, 1, 1}};
```

Output: 9

Your Task:

You don't need to read input or print anything. Your task is to complete the function **maxArea()** which takes the 2D array of booleans **mat**, **r** and **c** as parameters and returns an integer denoting the answer.

Expected Time Complexity: $O(R*(R + C))$

Expected Auxiliary Space: $O(R*C)$

Constraints:

$1 \leq R, C \leq 10^3$

$0 \leq \text{mat}[i][j] \leq 1$

Largest rectangle of 1s with swapping of columns allowed

Hard Accuracy: **55.1%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a matrix **mat** of size **R*C** with 0 and 1s, find the largest rectangle of all 1s in the matrix. The rectangle can be formed by swapping any pair of columns of given matrix.

Example 1:**Input:**

R = 3, C = 5

```
mat[][] = {{0, 1, 0, 1, 0},  
           {0, 1, 0, 1, 1},  
           {1, 1, 0, 1, 0}};
```

Output: 6

Explanation: The largest rectangle's area is 6. The rectangle can be formed by swapping column 2 with 3. The matrix after swapping will be

```
0 0 1 1 0  
0 0 1 1 1  
1 0 1 1 0
```

Example 2:**Input:**

R = 4, C = 5

```
mat[][] = {{0, 1, 0, 1, 0},  
           {0, 1, 1, 1, 1},  
           {1, 1, 1, 0, 1},  
           {1, 1, 1, 1, 1}};
```

Output: 9

Your Task:

You don't need to read input or print anything. Your task is to complete the

function **maxArea()** which takes the 2D array of booleans **mat**, **r** and **c** as parameters and returns an integer denoting the answer.

Expected Time Complexity: $O(R*(R + C))$

Expected Auxiliary Space: $O(R*C)$

Constraints:

$1 \leq R, C \leq 10^3$

$0 \leq \text{mat}[i][j] \leq 1$

Largest rectangle of 1s with swapping of columns allowed

Hard Accuracy: **55.1%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a matrix **mat** of size **R*C** with 0 and 1s, find the largest rectangle of all 1s in the matrix. The rectangle can be formed by swapping any pair of columns of given matrix.

Example 1:

Input:

$R = 3, C = 5$

$\text{mat}[][] = \{\{0, 1, 0, 1, 0\},$
 $\{0, 1, 0, 1, 1\},$
 $\{1, 1, 0, 1, 0\}\};$

Output: 6

Explanation: The largest rectangle's area is 6. The rectangle can be formed by swapping column 2 with 3. The matrix after swapping will be

```
0 0 1 1 0
0 0 1 1 1
1 0 1 1 0
```

Example 2:**Input:**

R = 4, C = 5

```
mat[][] = {{0, 1, 0, 1, 0},
            {0, 1, 1, 1, 1},
            {1, 1, 1, 0, 1},
            {1, 1, 1, 1, 1}};
```

Output: 9

Your Task:

You don't need to read input or print anything. Your task is to complete the function `maxArea()` which takes the 2D array of booleans `mat`, `r` and `c` as parameters and returns an integer denoting the answer.

Expected Time Complexity: $O(R*(R + C))$

Expected Auxiliary Space: $O(R*C)$

Constraints:

$1 \leq R, C \leq 10^3$

$0 \leq \text{mat}[i][j] \leq 1$

Doctor Strange

Hard Accuracy: **28.14%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Kamar-taj is a place where "The Ancient One" trains people to protect earth from other dimensions.

The earth is protected by N sanctums, destroying any of it will lead to invasion on earth.

The sanctums are connected by M bridges.

Now, you being on dormammu's side, want to find the number of sanctum destroying which will disconnect the sanctums.

Example 1:

Input:

$N = 5, M = 5$

$\text{arr}[] = \{\{1,2\},\{1,3\},\{3,2\},\{3,4\},\{5,4\}\}$

Output : 2

Explanation:

1. Removing 1 will not make graph disconnected (2--3--4--5).

2. Removing 2 will also not make graph disconnected (1--3--4--5).

3. Removing 3 makes graph disconnected (1--2 and 4--5).

4. Removing 4 makes graph disconnected (1--2--3--1 and 5).

5. Removing 5 also doesn't make graph disconnected (3--1--2--3--4).

6. Therefore, there are two such vertices, 3 and 4, so the answer is 2.

Example 2:

Input :

```
N = 2, M = 1  
arr[] = {{1, 2}}  
Output : 0
```

Your Task:

This is a function problem. The input is already taken care of by the driver code. You only need to complete the function **doctorStrange()** that takes a number of nodes (**N**), a number of edges (**M**), a 2-D matrix that contains connection between nodes (**graph**), and return the number of sanctums when destroyed will disconnect other sanctums of Earth.

Expected Time Complexity: $O(N + M)$.

Expected Auxiliary Space: $O(N + M)$.

Constraints:

$1 \leq n \leq 30000$

$1 \leq m \leq 30000$

$1 \leq u, v \leq n$

Smallest Positive Integer that can not be represented as Sum

Hard Accuracy: **35.8%** Submissions: **33K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array of size **N**, find the smallest positive integer value **that is either not presented in the array or cannot be represented as a sum (coz sum means you are adding two or more elements)** of some elements from the array.

Example 1:

Input:

$N = 6$

`array[] = {1, 10, 3, 11, 6, 15}`

Output:

2

Explanation:

2 is the smallest integer value that cannot be represented as sum of elements from the array.

Example 2:

Input:

$N = 3$

`array[] = {1, 1, 1}`

Output:

4

Explanation:

1 is present in the array.

2 can be created by combining two 1s.

3 can be created by combining three 1s.

4 is the smallest integer value that cannot be represented as sum of elements from the array.

Your Task:

You don't need to read input or print anything. Complete the function **smallestpositive()** which takes the array and N as input parameters and returns the smallest positive integer value that cannot be represented as the sum of some elements from the array.

Expected Time Complexity: $O(N * \text{Log}(N))$

Expected Auxiliary Space: $O(1)$

Constraints:

$$1 \leq N \leq 10^6$$

$$1 \leq \text{array}[i] \leq 10^9$$

The array may contain duplicates.

Number of subsets with product less than k

Hard Accuracy: **28.76%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `arr[]` of `N` elements. Find the number of non-empty subsets whose product of elements is less than or equal to a given integer `K`.

Example 1:**Input:**

$$N = 4$$

$$\text{arr}[] = \{2, 4, 5, 3\}$$

$$K = 12$$

Output:

8

Explanation:

All possible subsets whose products are less than 12 are:

(2), (4), (5), (3), (2, 4), (2, 5), (2, 3), (4, 3)

Example 2:**Input:**

$$N = 3$$

```
arr[] = {9, 8, 3}
```

```
K = 2
```

Output:

```
0
```

Explanation:

There is no subsets with product less than or equal to 2.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **numOfSubsets()** which takes 2 integers N, and K, and an array arr of size N as input and returns the number of subsets with product less equal to K.

Expected Time Complexity: $O(N \cdot 2^{N/2})$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 30$

$1 \leq \text{arr}[i] \leq 10$

$1 \leq K \leq 10^6$

Minimum X (xor) A

Hard Accuracy: **51.79%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two integers **A** and **B**, the task is to find an integer **X** such that **(X XOR A)** is minimum possible and the count of set bit in **X** is equal to the count of set bits in **B**.

Example 1:

Input:

A = 3, B = 5

Output: 3

Explanation:

Binary(A) = Binary(3) = 011

Binary(B) = Binary(5) = 101

The XOR will be minimum when $x = 3$

i.e. $(3 \text{ XOR } 3) = 0$ and the number

of set bits in 3 is equal

to the number of set bits in 5.

Example 2:

Input:

A = 7, B = 12

Output: 6

Explanation:

$(7)_2 = 111$

$(12)_2 = 1100$

The XOR will be minimum when $x = 6$

i.e. $(6 \text{ XOR } 7) = 1$ and the number

of set bits in 6 is equal to the

number of set bits in 12.

Your task :

You don't need to read input or print anything. Your task is to complete the function `minVal()` that takes integer A and B as input and returns the value of X according to the question.

Expected Time Complexity : $O(\log N)$

Expected Auxiliary Space : $O(1)$

Constraints :

$0 \leq A, B \leq 10^9$

Minimum operations to convert array A to B

Hard Accuracy: **21.58%** Submissions: **12K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two Arrays `A[]` and `B[]` of length `N` and `M` respectively. Find the minimum number of **insertions** and **deletions** on the array `A[]`, required to make both the arrays identical.

Note: Array `B[]` is sorted and all its elements are distinct, operations can be performed at any index not necessarily at end.

Example 1:

Input:

`N = 5, M = 3`

`A[] = {1, 2, 5, 3, 1}`

`B[] = {1, 3, 5}`

Output:

4

Explanation:

We need to delete 2 and replace it with 3.
This costs 2 steps. Further, we will have to delete the last two elements from A to obtain an identical array to B. Overall, it results in 4 steps.

Example 2:**Input:**

$N = 2, M = 2$

$A[] = \{1, 4\}$

$B[] = \{1, 4\}$

Output :

0

Explanation:

Both the Arrays are already identical.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `minInsAndDel()` which takes two integers N and M, and two arrays A of size N and B of size M respectively as input and returns the minimum insertions and deletions required.

Expected Time Complexity: $O(N \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 10^5$

$1 \leq A[i], B[i] \leq 10^5$

Split Array Largest Sum

Hard Accuracy: **60.13%** Submissions: **11K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `arr[]` of `N` elements and a number `K`. Split the given array into `K` subarrays such that the maximum subarray sum achievable out of `K` subarrays formed is minimum possible. Find that possible subarray sum.

Example 1:

Input:

`N = 4, K = 3`

`arr[] = {1, 2, 3, 4}`

Output: 4

Explanation:

Optimal Split is {1, 2}, {3}, {4}.

Maximum sum of all subarrays is 4,
which is minimum possible for 3 splits.

Example 2:

Input:

`N = 3, K = 2`

`A[] = {1, 1, 2}`

Output:

2

Explanation:

Splitting the array as {1,1} and {2} is optimal.

This results in a maximum sum subarray of 2.

Your Task:

The task is to complete the function `splitArray()` which returns the maximum sum subarray after splitting the array into K subarrays such that maximum sum subarray is minimum possible.

Constraints:

$$1 \leq N \leq 10^5$$

$$1 \leq K \leq N$$

$$1 \leq \text{arr}[i] \leq 10^4$$

Expected Time Complexity: $O(N \cdot \log(\text{sum}(\text{arr})))$.

Expected Auxiliary Space: $O(1)$.

Count the number of subarrays

Hard Accuracy: **37.85%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `A[]` of N integers and a range(L, R). The task is to find the number of subarrays having sum in the range L to R (inclusive).

Example 1:**Input:**

N = 3, L = 3, R = 8

A[] = {1, 4, 6}

Output:

3

Explanation:

The subarrays are [1,4], [4] and [6]

Example 2:**Input:**

$N = 4, L = 4, R = 13$

$A[] = \{2, 3, 5, 8\}$

Output:

6

Explanation:

The subarrays are [2,3], [2,3,5],
[3,5],[5], [5,8] and [8]

Your Task:

You don't need to read input or print anything. Complete the function `countSubarray()` which takes the integer N , the array $A[]$, the integer L and the integer R as input parameters and returns the number of subarrays.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq N \leq 10^6$

$1 \leq A[] \leq 10^9$

$1 \leq L \leq R \leq 10^{15}$

Burst Balloons

Hard Accuracy: **60.5%** Submissions: **281+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given N balloons, indexed from 0 to $n - 1$. Each balloon is painted with a number on it represented by an array **arr**. You are asked to burst all the balloons.

If you burst the **i**th balloon,, you will get $\text{arr}[i - 1] * \text{arr}[i] * \text{arr}[i + 1]$ coins. If $i - 1$, or $i + 1$ goes out of bounds of the array, consider it as if there is a balloon with a **1** painted on it. Return the **maximum** coins you can collect by bursting the balloons wisely.

Example 1:**Input:**

$N = 4$

$\text{arr}[] = \{3, 1, 5, 8\}$

Output: 167

Explanation:

$\text{arr}[] = \{3, 1, 5, 8\} \rightarrow \{3, 5, 8\} \rightarrow \{3, 8\} \rightarrow \{8\} \rightarrow \{\}$

$\text{coins} = 3 * 1 * 5, + 3 * 5 * 8 + 1 * 3 * 8 + 1 * 8 * 1 = 167$

Example 2:**Input:**

$N = 2$

$\text{arr}[] = \{1, 10\}$

Output: 20

Your Task:

You don't need to read input or print anything. Your task is to complete the function **maxCoins()** which takes the array of integers **arr** and **N** as parameters and returns the maximum coin you can collect.

Expected Time Complexity: $O(N*N*N)$

Expected Auxiliary Space: $O(N*N)$

Constraints:

$$1 \leq N \leq 300$$

$$0 \leq \text{arr}[i] \leq 100$$

Candy**Hard** Accuracy: **40.49%** Submissions: **220+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

There are **N** children standing in a line. Each child is assigned a rating value given in the integer array **ratings**.

You are giving candies to these children subjected to the following requirements:

- Each child must have atleast one candy.
- Children with a higher rating get more candies than neighbors.

Return the **minimum** number of candies you need to have to distribute the candies to the children.

Example 1:**Input:**

$$N = 3$$

$$\text{ratings}[] = \{1, 0, 2\}$$

Output: 5**Explanation:**

You can allocate to the first, second and third child with 2, 1, 2 candies respectively.

Example 2:**Input:**

$N = 3$

`ratings [] = {1, 2, 2}`

Output: 4**Explanation:**

You can allocate to the first, second and third child with 1, 2, 1 candies respectively.

The third child gets 1 candy because it satisfies the above two conditions.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `minCandy()` which takes the interger N and integer array `ratings[]` as parameters and returns the **minimum** number of candies you need to have to distribute the candies to the children.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 5 \cdot 10^4$

$0 \leq \text{ratings}_i \leq 10^5$

Candy

Hard Accuracy: **40.49%** Submissions: **220+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

There are N children standing in a line. Each child is assigned a rating value given in the integer array `ratings`.

You are giving candies to these children subjected to the following requirements:

- Each child must have atleast one candy.
- Children with a higher rating get more candies than neighbors.

Return the **minimum** number of candies you need to have to distribute the candies to the children.

Example 1:**Input:**

N = 3

ratings [] = {1, 0, 2}

Output: 5

Explanation:

You can allocate to the first, second and third child with 2, 1, 2 candies respectively.

Example 2:**Input:**

N = 3

ratings [] = {1, 2, 2}

Output: 4

Explanation:

You can allocate to the first, second and third child with 1, 2, 1 candies respectively.

The third child gets 1 candy because it statisfies the above two conditions.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **minCandy()** which takes the interger N and integer array **ratings[]** as

parameters and returns the **minimum** number of candies you need to have to distribute the candies to the children.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 5 \cdot 10^4$

$0 \leq \text{ratings}_i \leq 10^5$

Minimum X (xor) A

Hard Accuracy: **0.0%** Submissions: **0** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two integers **A** and **B**, the task is to find an integer **X** such that **(X XOR A)** is minimum possible and the count of set bit in **X** is equal to the count of set bits in **B**.

Example 1:

Input:

A = 3, B = 5

Output: 3

Explanation:

Binary(A) = Binary(3) = 011

Binary(B) = Binary(5) = 101

The XOR will be minimum when $x = 3$

i.e. $(3 \text{ XOR } 3) = 0$ and the number of set bits in 3 is equal

to the number of set bits in 5.

Example 2:**Input:**

A = 7, B = 12

Output: 6

Explanation:

$(7)_2 = 111$

$(12)_2 = 1100$

The XOR will be minimum when $x = 6$

i.e. $(6 \text{ XOR } 7) = 1$ and the number of set bits in 6 is equal to the number of set bits in 12.

Your task :

You don't need to read input or print anything. Your task is to complete the function `minVal()` that takes integer A and B as input and returns the value of X according to the question.

Expected Time Complexity : $O(\log N)$

Expected Auxiliary Space : $O(1)$

Constraints :

$1 \leq A, B \leq 10^9$

Allocate minimum number of pages

Hard Accuracy: **35.51%** Submissions: **107K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given N number of books. Every i th book has A_i number of pages.

You have to allocate contiguous books to M number of students. There can be many ways or permutations to do so. In each permutation, one of the M students will be allocated the maximum number of pages. Out of all these permutations, the task is to find that particular permutation in which the maximum number of pages allocated to a student is the minimum of those in all the other permutations and print this minimum value.

Each book will be allocated to exactly one student. Each student has to be allocated at least one book.

Note: Return -1 if a valid assignment is not possible, and allotment should be in contiguous order (see the explanation for better understanding).

Example 1:

Input:

$N = 4$

$A[] = \{12, 34, 67, 90\}$

$M = 2$

Output: 113

Explanation: Allocation can be done in

following ways: $\{12\}$ and $\{34, 67, 90\}$

Maximum Pages = 191 $\{12, 34\}$ and $\{67, 90\}$

Maximum Pages = 157 $\{12, 34, 67\}$ and $\{90\}$

Maximum Pages = 113. Therefore, the minimum

of these cases is 113, which is selected as the output.

Example 2:**Input:**

$N = 3$

$A[] = \{15, 17, 20\}$

$M = 2$

Output: 32

Explanation: Allocation is done as $\{15, 17\}$ and $\{20\}$

Your Task:

You don't need to read input or print anything. Your task is to complete the function `findPages()` which takes 2 Integers N , and m and an array $A[]$ of length N as input and returns the expected answer.

Expected Time Complexity: $O(N \log N)$

Expected Auxilliary Space: $O(1)$

Constraints:

$1 \leq N \leq 10^5$

$1 \leq A[i] \leq 10^6$

$1 \leq M \leq 10^5$

The Painter's Partition Problem-II

Hard Accuracy: **27.52%** Submissions: **55K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Dilpreet wants to paint his dog's home that has n boards with different lengths. The length of i^{th} board is given by $\text{arr}[i]$ where $\text{arr}[]$ is an array of n integers. He hired k painters for this work and each painter takes **1 unit time to paint 1 unit of the board**.

The problem is to find the minimum time to get this job done if all painters start together with the constraint that any painter will only paint continuous boards, say boards numbered $\{2,3,4\}$ or only board $\{1\}$ or nothing but not boards $\{2,4,5\}$.

Example 1:

Input:

$n = 5$

$k = 3$

$\text{arr}[] = \{5,10,30,20,15\}$

Output: 35

Explanation: The most optimal way will be:

Painter 1 allocation : $\{5,10\}$

Painter 2 allocation : $\{30\}$

Painter 3 allocation : $\{20,15\}$

Job will be done when all painters finish

i.e. at time = $\max(5+10, 30, 20+15) = 35$

Example 2:

Input:

$n = 4$

$k = 2$

$\text{arr}[] = \{10,20,30,40\}$

Output: 60

Explanation: The most optimal way to paint:

Painter 1 allocation : {10,20,30}

Painter 2 allocation : {40}

Job will be complete at time = 60

Your task:

Your task is to complete the function **minTime()** which takes the integers **n** and **k** and the array **arr[]** as input and returns the minimum time required to paint all partitions.

Expected Time Complexity: $O(n \log m)$, m = sum of all boards' length

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq n \leq 10^5$

$1 \leq k \leq 10^5$

$1 \leq \text{arr}[i] \leq 10^5$

Alien Dictionary

Hard Accuracy: **47.81%** Submissions: **50K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a sorted dictionary of an alien language having N words and k starting alphabets of standard dictionary. Find the order of characters in the alien language.

Note: Many orders may be possible for a particular test case, thus you may return

any valid order and output will be 1 if the order of string returned by the function is correct else 0 denoting incorrect string returned.

Example 1:**Input:**

N = 5, K = 4

dict = {"baa", "abcd", "abca", "cab", "cad"}

Output:

1

Explanation:

Here order of characters is

'b', 'd', 'a', 'c' Note that words are sorted

and in the given language "baa" comes before

"abcd", therefore 'b' is before 'a' in output.

Similarly we can find other orders.

Example 2:**Input:**

N = 3, K = 3

dict = {"caa", "aaa", "aab"}

Output:

1

Explanation:

Here order of characters is

'c', 'a', 'b' Note that words are sorted

and in the given language "caa" comes before

"aaa", therefore 'c' is before 'a' in output.

Similarly we can find other orders.

Your Task:

You don't need to read or print anything. Your task is to complete the function `findOrder()` which takes the string array `dict[]`, its size `N` and the integer `K` as input parameter and returns a string denoting the order of characters in the alien language.

Expected Time Complexity: $O(N * |S| + K)$, where $|S|$ denotes maximum length.

Expected Space Complexity: $O(K)$

Constraints:

$1 \leq N, M \leq 300$

$1 \leq K \leq 26$

$1 \leq \text{Length of words} \leq 50$

Merge Without Extra Space

Hard Accuracy: **32.01%** Submissions: **141K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two sorted arrays `arr1[]` and `arr2[]` of sizes `n` and `m` in non-decreasing order. Merge them in sorted order without using any extra space. Modify `arr1` so that it contains the first `N` elements and modify `arr2` so that it contains the last `M` elements.

Example 1:

Input:

```
n = 4, arr1[] = [1 3 5 7]
m = 5, arr2[] = [0 2 6 8 9]
```

Output:

```
arr1[] = [0 1 2 3]
arr2[] = [5 6 7 8 9]
```

Explanation:

After merging the two
non-decreasing arrays, we get,
0 1 2 3 5 6 7 8 9.

Example 2:**Input:**

```
n = 2, arr1[] = [10, 12]
m = 3, arr2[] = [5 18 20]
```

Output:

```
arr1[] = [5 10]
arr2[] = [12 18 20]
```

Explanation:

After merging two sorted arrays
we get 5 10 12 18 20.

Your Task:

You don't need to read input or print anything. You only need to complete the function **merge()** that takes arr1, arr2, n and m as input parameters and modifies them in-place so that they look like the sorted merged array when concatenated.

Expected Time Complexity: $O((n+m) \log(n+m))$

Expected Auxilliary Space: $O(1)$

Constraints: $1 \leq n, m \leq 10^5$ $0 \leq \text{arr1}_i, \text{arr2}_i \leq 10^7$ **Smallest window in a string containing all the characters of another string****Hard** Accuracy: **30.19%** Submissions: **96K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two strings **S** and **P**. Find the smallest window in the string **S** consisting of all the characters(**including duplicates**) of the string **P**. Return **"-1"** in case there is no such window present. In case there are multiple such windows of same length, return the one with the least starting index.

Example 1:**Input:****S** = "timetopractice"**P** = "toc"**Output:**

toprac

Explanation: "toprac" is the smallest substring in which "toc" can be found.**Example 2:****Input:****S** = "zoomlazapzo"**P** = "oza"**Output:**

apzo

Explanation: "apzo" is the smallest substring in which "oza" can be found.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **smallestWindow()** which takes two string **S** and **P** as input parameters and returns the smallest window in string **S** having all the characters of the string **P**. In case there are multiple such windows of same length, return the one with the least starting index.

Expected Time Complexity: $O(|S|)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq |S|, |P| \leq 10^5$

Joey doesn't share food

Hard Accuracy: **21.73%** Submissions: **497+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Every friends fan know that joey loves food and monica loves to cook. So, on a occasion of thanksgiving monica made n types of food containing exactly 6 ingredients each. Monica is an excellent cook so she can cook food by adding any ingredient at any order. All ingredients contains protein on the scale of 1 to 10^6 . Now, Chandler invented the fun game where everyone needs to find the successive protein rate in all n food of the ingredient on the top (6th ingredient is on top). Ross being the most curious wants to finish this game before dinner, so he wants your help to complete the task.

Example 1:

Input: nums = {{1,2,3,4,5,6},
{8,9,10,11,12,13}}

Output: 1

Explanation: No matter how the arrangement is the answer will be 1.

Example 2:

Input: nums = {{1,2,3,4,5,6},
{2,3,4,5,6,7}, {3,4,5,6,7,10},
{4,5,6,7,8,9}}

Output: 4

Explanation: If ingredients are arranged in any order and let say if don't rearrange it, ans will be 2.

As 6 7 or 10 9. These are the successive protein rate existing.

If we arrange it in order:

1 2 3 4 5 6

2 3 4 5 6 7

3 4 6 7 10 5

4 5 6 7 9 8

Now the answer will be 4.

As 6 7 5 8, so these protein rates are successive

(5 6 7 8).

Your Task: You don't need to read or print anything. Your task is to complete the function **MaximumLength()** which takes details of each food i.e nums as input

parameter and returns the length of largest chain formed.

Expected Time Complexity: $O(n^2)$

Expected Space Complexity: $O(n)$

Constraints:

$1 \leq n \leq 100$

$1 \leq \text{protein scale} \leq 10^6$

Joey doesn't share food

Hard Accuracy: **21.73%** Submissions: **497+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Every friends fan know that joey loves food and monica loves to cook. So, on a occassion of thanksgiving monica made n types of food containing exactly 6 ingredients each. Monica is an excellent cook so she can cook food by adding any ingredient at any order. All ingredients contains protein on the scale of 1 to 10^6 . Now, Chandler invented the fun game where everyone needs to find the successive protein rate in all n food of the ingredient on the top(6th ingredient is on top). Ross being the most curious wants to finish this game before dinner, so he wants your help to complete the task.

Example 1:

Input: nums = {{1,2,3,4,5,6},
{8,9,10,11,12,13}}

Output: 1

Explanation: No matter how the arrangement is the answer will be 1.

Example 2:

Input: nums = {{1,2,3,4,5,6},
{2,3,4,5,6,7}, {3,4,5,6,7,10},
{4,5,6,7,8,9}}

Output: 4

Explanation: If ingredients are arranged in any order and let say if don't rearrange it, ans will be 2.

As 6 7 or 10 9. These are the successive protein rate existing.

If we arrange it in order:

1 2 3 4 5 6

2 3 4 5 6 7

3 4 6 7 10 5

4 5 6 7 9 8

Now the answer will be 4.

As 6 7 5 8, so these protein rates are successive (5 6 7 8).

Your Task: You don't need to read or print anything. Your task is to complete the function **MaximumLength()** which takes details of each food i.e nums as input parameter and returns the length of largest chain formed.

Expected Time Complexity: $O(n^2)$

Expected Space Complexity: $O(n)$

Constraints:

$1 \leq n \leq 100$

$1 \leq \text{protein scale} \leq 10^6$

Max rectangle

Hard Accuracy: **36.43%** Submissions: **65K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a binary matrix **M** of size **n X m**. Find the maximum area of a rectangle formed only of **1s** in the given matrix.

Example 1:

Input:

$n = 4, m = 4$

$M[][] = \{\{0\ 1\ 1\ 0\},$

$\{1\ 1\ 1\ 1\},$

$\{1\ 1\ 1\ 1\},$

$\{1\ 1\ 0\ 0\}\}$

Output: 8

Explanation: For the above test case the matrix will look like

0 1 1 0

1 1 1 1

1 1 1 1

```
1 1 0 0
```

the max size rectangle is

```
1 1 1 1
```

```
1 1 1 1
```

and area is $4 * 2 = 8$.

Your Task:

Your task is to complete the function **maxArea** which returns the maximum size rectangle area in a binary-sub-matrix with all 1's. The function takes 3 arguments the first argument is the Matrix `M[][]` and the next two are two integers `n` and `m` which denotes the size of the matrix `M`.

Expected Time Complexity : $O(n*m)$

Expected Auxiliary Space : $O(m)$

Constraints:

$1 \leq n, m \leq 1000$

$0 \leq M[i][j] \leq 1$

Note: The **Input/Output** format and **Example** given are used for system's internal purpose, and should be used by a user for **Expected Output** only. As it is a function problem, hence a user should not read any input from stdin/console. The task is to complete the function specified, and not to write the full code.

Solve the Sudoku

Hard Accuracy: **37.98%** Submissions: **57K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an incomplete Sudoku configuration in terms of a 9×9 2-D square matrix (`grid[][]`), the task to find a solved Sudoku. For simplicity, you may assume that

there will be only one unique solution.

Sample Sudoku for you to get the logic for its solution:

Example 1:

Input:

```
grid[][] =  
[[3 0 6 5 0 8 4 0 0],  
[5 2 0 0 0 0 0 0 0],  
[0 8 7 0 0 0 0 3 1],  
[0 0 3 0 1 0 0 8 0],  
[9 0 0 8 6 3 0 0 5],  
[0 5 0 0 9 0 6 0 0],  
[1 3 0 0 0 0 2 5 0],  
[0 0 0 0 0 0 0 7 4],  
[0 0 5 2 0 6 3 0 0]]
```

Output:

```
3 1 6 5 7 8 4 9 2  
5 2 9 1 3 4 7 6 8  
4 8 7 6 2 9 5 3 1  
2 6 3 4 1 5 9 8 7  
9 7 4 8 6 3 1 2 5  
8 5 1 7 9 2 6 4 3  
1 3 8 9 4 7 2 5 6  
6 9 2 3 5 1 8 7 4  
7 4 5 2 8 6 3 1 9
```

Your Task:

You need to complete the two functions:

SolveSudoku(): Takes a grid as its argument and returns true if a solution is possible and false if it is not.

printGrid(): Takes a grid as its argument and prints the 81 numbers of the solved Sudoku in a single line with space separation.

NOTE: Do not give a new line character after printing the grid. It has already been taken care of in the Driver Code.

Expected Time Complexity: $O(9^{N*N})$.

Expected Auxiliary Space: $O(N*N)$.

Constraints:

$$0 \leq \text{grid}[i][j] \leq 9$$

Matrix Chain Multiplication

Hard Accuracy: **49.64%** Submissions: **71K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a sequence of matrices, find the most efficient way to multiply these matrices together. The efficient way is the one that involves the least number of multiplications.

The dimensions of the matrices are given in an array **arr[]** of size **N** (such that **N** = number of matrices + 1) where the **ith** matrix has the dimensions (**arr[i-1] x arr[i]**).

Example 1:

Input: N = 5

arr = {40, 20, 30, 10, 30}

Output: 26000

Explanation: There are 4 matrices of dimension 40x20, 20x30, 30x10, 10x30. Say the matrices are named as A, B, C, D. Out of all possible combinations, the most efficient way is $(A*(B*C))*D$.

The number of operations are -

$20*30*10 + 40*20*10 + 40*10*30 = 26000$.

Example 2:

Input: N = 4

arr = {10, 30, 5, 60}

Output: 4500

Explanation: The matrices have dimensions 10*30, 30*5, 5*60. Say the matrices are A, B and C. Out of all possible combinations, the most efficient way is $(A*B)*C$. The number of multiplications are -

$10*30*5 + 10*5*60 = 4500$.

Your Task:

You do not need to take input or print anything. Your task is to complete the function **matrixMultiplication()** which takes the value N and the array **arr[]** as input parameters and returns the minimum number of multiplication operations needed to be performed.

Expected Time Complexity: $O(N^3)$

Expected Auxiliary Space: $O(N^2)$

Constraints:

$$2 \leq N \leq 100$$

$$1 \leq \text{arr}[i] \leq 500$$

Generalised Fibonacci numbers

Hard Accuracy: **13.7%** Submissions: **4K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Consider the generalized Fibonacci number G , which is dependent on a , b and c as follows :-

$$G(1) = 1, G(2) = 1. G(n) = aG(n-1) + bG(n-2) + c.$$

Your task is to calculate $G(n) \% m$ for given values of n and m .

Example 1:

Input:

$$a = 3, b = 3, c = 3, n = 3, m = 5$$

Output:

4

Explanation:

$$G(3) = 3 * G(2) + 3 * G(1) + 3 = 9 \% 5 = 4$$

Example 2:

Input:

$a = 2, b = 2, c = 2, n = 4, m = 100$

Output:

16

Explanation:

$$G(3) = 2 * G(2) + 2 * G(1) + 2 = 6$$

$$G(4) = 2 * G(3) + 2 * G(2) + 2 = 16 \% 100 = 16$$

Your Task:

You don't need to read input or print anything. Your task is to complete the function **genFibNum()** which takes 5 Integers a, b, c, n, and m as input and returns $G(n) \% m$.

Expected Time Complexity: $O(\log n)$

Expected Auxiliary Space: $O(1)$

Constraints:

$$1 \leq a, b, c, n, m \leq 10^9 + 7$$

Brackets in Matrix Chain Multiplication

Hard Accuracy: **62.74%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array $p[]$ of length n used to denote the dimensions of a series of matrices such that dimension of i 'th matrix is $p[i] * p[i+1]$. There are a total of $n-1$ matrices. Find the most efficient way to multiply these matrices together. The problem is not actually to perform the multiplications, but merely to decide in which order to perform the multiplications such that you need to perform minimum number of multiplications. There are many options to multiply a chain of matrices because matrix multiplication is associative i.e. no matter how one parenthesize the product, the result will be the same.

Example 1:**Input:** $n = 5$ $p[] = \{1, 2, 3, 4, 5\}$ **Output:** $((AB)C)D$

Explanation: The total number of multiplications are $(1*2*3) + (1*3*4) + (1*4*5) = 6 + 12 + 20 = 38$.

Example 2:**Input:** $n = 3$ $p = \{3, 3, 3\}$ **Output:** (AB)

Explanation: The total number of multiplications are $(3*3*3) = 27$.

Your Task:

You do not need to read input or print anything. Your task is to complete the function `matrixChainOrder()` which takes `n` and `p[]` as input parameters and returns the string with the proper order of parenthesis for `n-1` matrices. Use uppercase alphabets to denote each matrix.

Expected Time Complexity: $O(n^3)$

Expected Auxiliary Space: $O(n^2)$

Constraints:

$$2 \leq n \leq 26$$

$$1 \leq p[i] \leq 500$$

Rohan's Love for Matrix

Hard Accuracy: **25.65%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Rohan has a special love for the matrices especially for the first element of the matrix. Being good at Mathematics, he also loves to solve the different problem on the matrices. So one day he started to multiply the matrix with the original matrix. The elements of the original matrix are given by $a_{00}=1$, $a_{01}=1$, $a_{10}=1$, $a_{11}=0$. Given the power of the matrix, n calculate the a^n and return the a_{10} element mod 1000000007.

Example 1:

Input: `n = 3`

Output: 2

Explanation: Take the cube of the original matrix
i.e a^3 and the first element(a_{10}) is 2.

Example 2:

Input: $n = 4$

Output: 3

Explanation: Take the cube of the original matrix
i.e a^4 and the first element(a_{10}) is 3.

Your Task:

You dont need to read input or print anything. Complete the function `firstElement()` which takes n as input parameter and returns the a_{10} element mod 1000000007.

Expected Time Complexity: $O(n)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq n \leq 10^6$

Queries on a Matrix

Hard Accuracy: **68.34%** Submissions: **11K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given a matrix of dimension $n*n$. All the cells are initially, zero. You are given Q queries, which contains 4 integers $a\ b\ c\ d$ where (a,b) is the TOP LEFT

cell and (c,d) is the Bottom Right cell of a submatrix. Now, all the cells of this submatrix have to be incremented by one. After all the Q queries have been performed. Your task is to find the final resulting Matrix.

Note : Zero-Based Indexing is used for cells of the matrix.

Example 1:

Input: n = 6, q = 6,

Queries = {

{4,0,5,3},

{0,0,3,4},

{1,2,1,2},

{1,1,2,3},

{0,0,3,1},

{1,0,2,4}}.

Output:

2 2 1 1 1 0

3 4 4 3 2 0

3 4 3 3 2 0

2 2 1 1 1 0

1 1 1 1 0 0

1 1 1 1 0 0

Explanation:After incrementing all the sub-matrices of given queries we will get the final output.

Example 2:

Input: $n = 4, q = 2,$

Queries = {

{0,0,3,3},

{0,0,2,2}}.

Output:

2 2 2 1

2 2 2 1

2 2 2 1

1 1 1 1

Explanation:After incrementing all the sub-matrices of given queries we will get the final output.

Your Task:

You don't need to read or print anything. Your task is to complete the function `solveQueries()` which takes n and Queries and input parameter and returns a matrix after performing all the queries.

Expected Time Complexity: $O(n^2)$

Expected Space Complexity: $O(n^2)$

Constraints:

$1 \leq n \leq 1000$

$0 \leq a \leq c < n$

$0 \leq b \leq d < n$

$1 \leq \text{No. of Queries} \leq 1000$

Adventure in a Maze

Hard Accuracy: **32.75%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You have got a maze, which is a $n \times n$ Grid. Every cell of the maze contains these numbers 1, 2 or 3.

If it contains 1 : means we can go Right from that cell only.

If it contains 2 : means we can go Down from that cell only.

If it contains 3 : means we can go Right and Down to both paths from that cell.

We cant go out of the maze at any time.

Initially, You are on the Top Left Corner of The maze(Entry). And, You need to go to the Bottom Right Corner of the Maze(Exit).

You need to find the total number of paths from Entry to Exit Point.

There may be many paths but you need to select that path which contains the maximum number of Adventure.

The Adventure on a path is calculated by taking the sum of all the cell values that lies in the path.

Example 1:

Input: matrix = {{1,1,3,2,1},{3,2,2,1,2},
{1,3,3,1,3},{1,2,3,1,2},{1,1,1,3,1}}

Output: {4,18}

Explanation: There are total 4 Paths Available out of which The Max Adventure is 18. Total possible Adventures are 18,17,17,16. Of these 18 is the maximum.

Your Task:

You don't need to read or print anything. Your task is to complete the function **FindWays()** which takes matrix as input parameter and returns a list containing total number of ways to reach at (n, n) modulo $10^9 + 7$ and maximum number of Adventure.

Expected Time Complexity: $O(n^2)$

Expected Space Complexity: $O(n^2)$

Constraints:

$1 \leq n \leq 100$

Let's Play!!!

Hard Accuracy: **34.49%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Let's play a game!! Given a matrix `mat[][]` with $n \times m$ elements. Your task is to check that matrix is Super Similar or not. To perform this task you have to follow these Rules: Firstly all even index rows to be Rotated left and odd index rows to right, And Rotation is done X times (Index starting from zero). Secondly, After all the Rotations check if the initial and the final Matrix are same Return 1 else 0.

Example 1:

Input: $n = 2, m = 2$

`mat = {{1, 2},`


```
{5, 6}}
```

```
x = 1
```

Output: 0

Explanation: Matrix after rotation:

```
mat = {{ 2, 1}
```

```
      { 6, 5}}
```

After one rotation mat is

not same as the previous one.

Example 2:

Input: n = 2, m = 4

```
mat = {{1, 2, 1, 2},
```

```
      {2, 1, 2, 1}}
```

```
x = 2
```

Output: 1

Explanation: After two rotation mat is same as the previous one.

Your Task:

You do not need to read input or print anything. Your task is to complete the function `isSuperSimilar()` which takes n, m, x and the matrix as input parameter and returns 1 if the initial and the final Matrix are same else returns 0.

Expected Time Complexity: $O(n*m)$

Expected Auxiliary Space: $O(n*m)$

Constraints:

$$1 \leq n, m \leq 30$$

$$1 \leq \text{mat}[i][j] \leq 100$$

$$1 \leq x \leq 20$$

Find duplicate rows in a binary matrix**Hard** Accuracy: **62.89%** Submissions: **3K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a boolean matrix of size RxC where each cell contains either 0 or 1, find the row numbers of row (0-based) which already exists or are repeated.

Example 1:**Input:**

R = 2, C = 2

matrix[][] = {{1, 0},
 {1, 0}}

Output:

1

Explanation:

Row 1 is duplicate of Row 0.

Example 2:**Input:**

R = 4, C = 3

matrix[][] = {{ 1, 0, 0},
 { 1, 0, 0},

```
{ 1, 0, 0},  
{ 0, 0, 0}}
```

Output:

1 2

Explanation:

Row 1 and Row 2 are duplicates of Row 0.

Your Task:

You dont need to read input or print anything. Complete the function **repeatedRows()** that takes the matrix as input parameter and returns a list of row numbers which are duplicate rows.

Expected Time Complexity: $O(R * C)$

Expected Auxiliary Space: $O(R*C)$

Constraints:

$1 \leq R \leq 1000$

$1 \leq C \leq 20$

$0 \leq \text{matrix}[i][j] \leq 1$

Find number of closed islands

Hard Accuracy: **56.75%** Submissions: **2K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a [binary matrix](#) **mat[][]** of dimensions **NxM** such that 1 denotes land and 0 denotes water. Find the number of closed islands in the given matrix.

A closed island is known as the group of **1s** that is surrounded by only **0s** on all the four sides (excluding diagonals). If any **1** is at the edges of the given matrix then it is not considered as the part of the connected island as it is not surrounded by all **0's**.

Example 1:

Input:

N = 5, M = 8

```
mat[][] = {{0, 0, 0, 0, 0, 0, 0, 1},
            {0, 1, 1, 1, 1, 0, 0, 1},
            {0, 1, 0, 1, 0, 0, 0, 1},
            {0, 1, 1, 1, 1, 0, 1, 0},
            {0, 0, 0, 0, 0, 0, 0, 1}}
```

Output:

2

Explanation:

```
mat[][] = {{0, 0, 0, 0, 0, 0, 0, 1},
            {0, 1, 1, 1, 1, 0, 0, 1},
            {0, 1, 0, 1, 0, 0, 0, 1},
            {0, 1, 1, 1, 1, 0, 1, 0},
            {0, 0, 0, 0, 0, 0, 0, 1}}
```

There are 2 closed islands.

The islands in dark are closed because they are completely surrounded by 0s (water).

There are two more islands in the last column of the matrix, but they are not completely surrounded by 0s.

Hence they are not closed islands.

Example 2:

Input:

N = 3, M = 3

mat[][] = {{1, 0, 0},
 {0, 1, 0},
 {0, 0, 1}}

Output:

1

Your task:

You don't need to read input or print anything. Your task is to complete the function **closedIslands()** which takes two integers N and M, and a 2D binary matrix mat as input parameters and returns the number of closed islands.

Expected Time Complexity: $O(N \cdot M)$

Expected Auxiliary Space: $O(N \cdot M)$

Constraints:

$1 \leq N, M \leq 500$

Shortest Path by Removing K walls

Hard Accuracy: **50.4%** Submissions: **15K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a 2-D binary **matrix** of size **n*m**, where 0 represents an empty space while 1 represents a wall you cannot walk through. You are also given an integer **k**. You can walk up, down, left, or right. Given that you can remove up to **k** walls, return the minimum number of steps to walk from the top left corner (0, 0) to the bottom right corner (n-1, m-1).

Note: If there is no way to walk from the top left corner to the bottom right corner, return -1.

Example 1:

Input: n = 3, m = 3, k = 1

mat = {{0, 0, 0},
 {0, 0, 1},
 {0, 1, 0}}

Output:

4

Explanation:

We can remove any one of the walls and reach the bottom in 4 moves.

Example 2:

Input:

n = 2, m = 2, k = 0

mat[] = {{0, 1},
 {1, 0}}

Output:

-1

Explanation:

There's no way of reaching the bottom corner without removing any walls.

Your Task:

The task is to complete the function `shortestPath()` which takes three integers n , m , and k and also a matrix of size $n*m$ as input and returns the minimum number of steps to walk from the top left corner to the bottom right corner.

Constraints:

$$1 \leq n, m \leq 50$$

$$0 \leq k \leq n*m$$

Top left and bottom right corners doesn't have 1

Expected Time Complexity: $O(n*m*k)$.

Expected Auxiliary Space: $O(n*m*k)$.

Longest Palindromic Substring in Linear Time

Hard Accuracy: **14.32%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string, find the longest substring which is palindrome in Linear time $O(N)$.

Input:

The first line of input contains an integer T denoting the no of test cases . Then T test cases follow. The only line of each test case contains a string.

Output:

For each test case print the Longest Palindromic Substring. If there are multiple such substrings of same length, print the one which appears first in the input string.

Constraints: $1 \leq T \leq 100$ $1 \leq N \leq 50$ **Example:****Input:**

2

babcbabcbaccba

forgeeksskeegfor

Output:

abcbabcba

geeksskeeg

Note:The **Input/Output** format and **Example** given are used for system's internal purpose, and should be used by a user for **Expected Output** only. As it is a function problem, hence a user should not read any input from stdin/console. The task is to complete the function specified, and not to write the full code.

Longest Palindromic Substring in Linear Time**Hard** Accuracy: **14.32%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string, find the longest substring which is palindrome in Linear time $O(N)$.

Input:

The first line of input contains an integer T denoting the no of test cases . Then T test cases follow. The only line of each test case contains a string.

Output:

For each test case print the Longest Palindromic Substring. If there are multiple such substrings of same length, print the one which appears first in the input string.

Constraints:

$1 \leq T \leq 100$

$1 \leq N \leq 50$

Example:**Input:**

2

babcbabcbaccba

forgeeksskeegfor

Output:

abcbabcba

geeksskeeg

Note:The **Input/Output** format and **Example** given are used for system's internal purpose, and should be used by a user for **Expected Output** only. As it is a function problem, hence a user should not read any input from stdin/console. The task is to complete the function specified, and not to write the full code.

Distinct palindromic substrings

Hard Accuracy: **39.92%** Submissions: **21K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string `str` of lowercase ASCII characters, Count the number of distinct continuous palindromic sub-strings which are present in the string `str`.

Example 1:

Input:

`str = "abaaa"`

Output:

5

Explanation : These are included in answer:

`"a", "aa", "aaa", "aba", "b"`

Example 2:

Input

`str = "geek"`

Output:

4

Explanation : Below are 4 palindrome sub-strings

`"e", "ee", "g", "k"`

Your Task:

You don't need to read input or print anything. Your task is to complete the function `palindromeSubStrs()` which takes the string `str` as input parameter and returns the total number of distinct continuous palindromic sub-strings in `str`.

Expected Time Complexity : $O(N^2 \log N)$

Expected Auxilliary Space : $O(N^2)$

Constraints:

$1 \leq N \leq 3 \cdot 10^3$, where N is the length of the string str.

Number of distinct subsequences

Hard Accuracy: **19.81%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string consisting of lower case English alphabets, the task is to find the number of distinct subsequences of the string

Note: Answer can be very large, so, output will be answer modulo 10^9+7

Example 1:

Input: s = "gfg"

Output: 7

Explanation: The seven distinct subsequences are "", "g", "f", "gf", "fg", "gg" and "gfg"

Example 2:

Input: s = "ggg"

Output: 4

Explanation: The four distinct subsequences are "", "g", "gg", "ggg"

Your task:

You do not need to read any input or print anything. The task is to complete the function **distinctSubsequences()**, which takes a string as input and returns an integer.

Expected Time Complexity: $O(|str|)$

Expected Auxiliary Space: $O(|str|)$

Constraints:

$1 \leq |s| \leq 10^5$

s contains lower case English alphabets

Wildcard string matching

Hard Accuracy: **23.88%** Submissions: **20K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two strings **wild** and **pattern** where wild string may contain wild card characters and pattern string is a normal string. Determine if the two strings match. The following are the allowed wild card characters in first string :-

* --> This character in string wild can be replaced by any sequence of characters, it can also be replaced by an empty string.

? --> This character in string wild can be replaced by any one character.

Example 1:

Input: wild = ge*ks

pattern = geeks

Output: Yes

Explanation: Replace the '*' in wild string with 'e' to obtain pattern "geeks".

Example 2:

Input: wild = ge?ks*

```
pattern = geeksforgeeks
```

Output: Yes

Explanation: Replace '?' and '*' in wild string with 'e' and 'forgeeks' respectively to obtain pattern "geeksforgeeks"

Your Task:

You don't need to read input or print anything. Your task is to complete the function `match()` which takes the string `wild` and `pattern` as input parameters and returns true if the string `wild` can be made equal to the string `pattern`, otherwise, returns false.

Expected Time Complexity: $O(\text{length of wild string} * \text{length of pattern string})$

Expected Auxiliary Space: $O(\text{length of wild string} * \text{length of pattern string})$

Constraints:

$1 \leq \text{length of the two string} \leq 10^3$

Longest valid Parentheses

Hard Accuracy: **26.13%** Submissions: **71K** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string `S` consisting of opening and closing parenthesis '(' and ')'. Find length of the longest valid parenthesis substring.

A parenthesis string is valid if:

- For every opening parenthesis, there is a closing parenthesis.
- Opening parenthesis must be closed in the correct order.

Example 1:

Input: S = (()

Output: 2

Explanation: The longest valid parenthesis substring is "()".

Example 2:

Input: S =)()())

Output: 4

Explanation: The longest valid parenthesis substring is "()()".

Your Task:

You do not need to read input or print anything. Your task is to complete the function **maxLength()** which takes string S as input parameter and returns the length of the maximum valid parenthesis substring.

Expected Time Complexity: $O(|S|)$

Expected Auxiliary Space: $O(|S|)$

Constraints:

$1 \leq |S| \leq 10^5$

Elixir of Life

Hard Accuracy: **58.42%** Submissions: **3K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Flamel is making the Elixir of Life but he is missing a secret ingredient, a set of contiguous plants (substring) from the Garden of Eden.

The garden consists of various plants represented by string S, where each letter

represents a different plant. But the prophecy has predicted that the correct set of plants required to make the potion will appear in the same contiguous pattern (substring) at the beginning of the forest (**prefix**), the end of the forest (**suffix**), and will also be the most frequent sequence present in the entire forest.

Identify the substring of plants required to make the elixir and find out the number of times it appears in the forest.

Example 1:

Input: S = "ababaaaab"

Output: 3

Explanation: Substring "ab" is a prefix, It is also a suffix and appears 3 times.

Example 2:

Input: S = "aaaa"

Output: 4

Explanation: Substring "aaaa" occurs 1 time, Substring "aaa" occurs 2 times, substring "aa" occurs 3 times, substring "a" occurs 4 times. All of them are proper prefixes and suffixes. But, "a" has the maximum frequency.

Example 3:

Input: S = "abcdef"

Output: 1

Your Task:

You don't need to read input or print anything. Complete the function **maxFrequency()** which takes string *S* as input parameter and returns the frequency of the most frequent substring of *S* which is also a prefix and suffix of the original string.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq |S| \leq 10^5$

Find all possible palindromic partitions of a String

Hard Accuracy: **62.13%** Submissions: **12K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a String *S*, Find all possible Palindromic partitions of the given String.

Example 1:

Input:

S = "geeks"

Output:

g e e k s

g e e k s

Explanation:

All possible palindromic partitions are printed.

Example 2:**Input:**`S = "madam"`**Output:**`m a d a m``m a d a m``madam`**Your Task:**

You don't need to read input or print anything. Your task is to complete the function `allPalindromicPerms()` which takes a String `S` as input parameter and returns a list of lists denoting all the possible palindromic partitions in the order of their appearance in the original string.

Expected Time Complexity: $O(N \cdot 2^N)$ **Expected Auxiliary Space:** $O(N^2)$, where `N` is the length of the String**Constraints:** $1 \leq |S| \leq 20$ **Longest substring to form a Palindrome****Hard** Accuracy: **50.56%** Submissions: **6K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string **S** which only contains lowercase alphabets. Find the length of the longest substring of **S** such that the characters in it can be rearranged to form a [palindrome](#).

Example 1:**Input:**

S = "aabe"

Output:

3

Explanation:

The substring "aab" can be rearranged to "aba" which is the longest palindrome possible for this String.

Example 2:**Input:**

S = "adbabd"

Output:

6

Explanation:

The whole string "adbabd" can be rearranged to form a palindromic substring. One possible arrangement is "abddba". Thus, output length of the string is 6.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **longestSubstring()** which takes a String **S** as input and returns the length of largest possible Palindrome.

Expected Time Complexity: $O(|S|*26)$

Expected Auxiliary Space: $O(|S|*26)$

Constraints:

$1 \leq |S| \leq 10^5$

Number of distinct words with K maximum contiguous vowels

Hard Accuracy: **48.53%** Submissions: **6K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Find the number of unique words consisting of lowercase alphabets only of length N that can be formed with at-most K contiguous vowels.

Example 1:

Input:

N = 2

K = 0

Output:

441

Explanation:

Total of 441 unique words are possible of length 2 that will have K(=0) vowels together, e.g. "bc", "cd", "df", etc are valid words while "ab" (with 1 vowel) is not a valid word.

Example 2:**Input:**

N = 1

K = 1

Output:

26

Explanation:

All the english alphabets including vowels and consonants; as atmost K(=1) vowel can be taken.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `kvowelwords()` which takes an Integer N, an integer K and returns the total number of words of size N with atmost K vowels. As the answer may be too large please return answer modulo 1000000007.

Expected Time Complexity: $O(N \cdot K)$

Expected Auxiliary Space: $O(N \cdot K)$

Constraints:

$1 \leq N \leq 1000$

$0 \leq K \leq N$

Search Query Auto Complete

Hard Accuracy: 16.67% Submissions: 204+ Points: 8

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Design a search query autocomplete system for a search engine.

The users will input a sentence (which may have multiple words and ends with special character '#').

For each character they type except '#', you need to return the top 3 previously entered and most frequently queried sentences that have prefix the same as the part of sentence already typed.

Here are the specific rules:

1. The frequency for a sentence is defined as the number of times a user typed the exactly same sentence before.
2. The returned top 3 sentences should be sorted by frequency (The first is the most frequent). If several sentences have the same frequency, you need to use ASCII-code order (smaller one appears first).
3. If less than 3 valid sentences exist, then just return as many as you can.
4. When the input is a special character, it means the sentence ends, and in this case, you need to return an empty list.

Your job is to implement the methods of the `AutoCompleteSystem`:

- `AutoCompleteSystem(String[] sentences, int[] times)`: This is the constructor. The input is previously used data. Sentences is a string array consists of previously typed sentences. Times is the corresponding times a sentence has been typed. Your system should record these historical sentences.

Now, the user wants to input a new sentence. The following function will provide the next character the user types:

- `String[] input(char c)`: The input `c` is the next character typed by the user. The character will only be lower-case letters ('a' to 'z'), blank space (' ') or a special character ('#'). Also, the previously typed sentence should be recorded in your system. The output an array will be the top 3 historical sentences that have prefix the same as the part of sentence already typed.

Example:

Operation:

```
AutoCompleteSystem(["i love you", "island",  
"ironman", "i love geeksforgeeks"], [5,3,2,2])
```

The system have already tracked down the following sentences and their corresponding times:

"i love you" : 5 times

"island" : 3 times

"ironman" : 2 times

"i love geeksforgeeks" : 2 times

Now, the user begins another search:

Operation: `input('i')`

Output:

```
["i love you", "island", "i love  
geeksforgeeks"]
```

Explanation:

There are four sentences that have prefix "i". Among them, "ironman" and "i love

geeksforgeeks" have same frequency. Since ' ' has ASCII code 32 and 'r' has ASCII code 114, "i love geeksforgeeks" should be in front of "ironman". Also we only need to output top 3 most frequent sentences, so "ironman" will be ignored.

Operation: input(' ')

Output: ["i love you", "i love geeksforgeeks"]

Explanation:

There are only two sentences that have prefix "i ".

Operation: input('a')

Output: []

Explanation:

There are no sentences that have prefix "i a"

Operation: input('#')

Output: []

Explanation:

The user finished the input, the sentence "i a" should be saved as a historical sentence in system. And the next input will be counted as a new search.

Your Task:

You don't need to take inputs or give outputs . You just have to complete the **input()** method and the **constructor**.

Expected Time Complexity: $O(n \cdot \max|L|)$, per input query where n represents the number of historical sentences in the system and L is the maximum length of the words.

Expected Time Complexity: $O(|\text{sentences}|)$, $|\text{sentences}|$ represents the total length of all historical sentences in the system.

Constraints:

- The input sentence will always start with a letter and end with '#', and at most one blank space will exist between two words.
- The number of complete sentences that to be searched won't exceed 10^2 .
- The length of each sentence including those in the historical data and query data won't exceed 10^2 .

IPL 2021 - Final

Hard Accuracy: **33.95%** Submissions: **14K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

IPL 2021 Finals are here and it is between the most successful team of the IPL Mumbai Indians and the team striving to grab their first trophy Royal Challengers Bangalore. Rohit Sharma, captain of the team Mumbai Indians has the most experience in IPL finals, he feels lucky if he solves a programming question before the IPL finals. So, he asked the team's head coach Mahela Jayawardene for a question. Question is, given a string S consisting only of opening and closing parenthesis 'ie '(' and ')', the task is to find out the length of the longest valid parentheses substring.

NOTE: The length of the smallest valid substring $()$ is 2.

Example 1:**Input:** S = "()("**Output:** 2**Explanation:** The longest valid substring is "()". Length = 2.**Example 2:****Input:** S = "()()("**Output:** 6**Explanation:** The longest valid substring is "()()". Length = 6.**Your Task:**

You don't need to read input or print anything. Complete the function **findMaxLen()** which takes **S** as input parameter and returns the max length.

Constraints:

$$1 \leq |S| \leq 10^5$$

Scrambled String**Hard** Accuracy: **100.0%** Submissions: **1** + Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two strings **S1** and **S2** of equal length, the task is to determine if **S2** is a scrambled form of **S1**.

Scrambled string: Given string **str**, we can represent it as a binary tree by partitioning it into two non-empty substrings recursively.

Note: Scrambled string is not the same as an Anagram.

Print "Yes" if S2 is a scrambled form of S1 otherwise print "No".

Example 1:

Input: S1="coder", S2="ocder"

Output: Yes

Explanation: ocder is a scrambled form of coder.

```

    ocred
   /  \
  oc  red
 /\  /\
o c re d
   /\
   r e

```

As "ocder" can represent it as a binary tree by partitioning it into two non-empty substrings.

Example 2:

Input: S1="abcde", S2="caebd"

Output: No

Explanation: caebd is not a scrambled form of abcde

Your Task:

You don't need to read input or print anything. You only need to complete the function `isScramble()` which takes two strings `S1` and `S2` as input and returns a boolean value.

Constraints:

- `S1.length = S2.length`
- `S1.length ≤ 31`
- `S1` and `S2` consist of lower-case English letters.

Find the String

Hard Accuracy: **59.55%** Submissions: **408+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two integer `N` and `K`. The task is to find the string `S` of minimum length such that it contains all possible strings of size `N` as a substring. The characters of the string can be from `0` to `K-1`.

Example 1:**Input:**

`N = 2, K = 2`

Output:

`00110`

Explanation:

There are 4 string possible of size `N=2`
which contains characters `0,..K-1`
(i.e `"00"`, `"01"`, `"10"`, `"11"`)

"00110" contains all possible string as a substring. It also has the minimum length

Example 2:**Input:**

N = 2, K = 3

Output:

0010211220

Your Task:

You don't need to read input or print anything. Complete the function **findString()** which takes the integer **N** and the integer **K** as input parameters and returns the string.

Note: In case of multiple answers, return any string of minimum length which satisfies above condition. The driver will print the length of the string. In case of wrong answer it will print -1.

Constraints:

$$1 \leq N \leq 4$$

$$1 \leq K \leq 10$$

$$1 \leq K^N \leq 4096$$

Minimum characters to be added at front to make string palindrome

Hard Accuracy: **46.79%** Submissions: **22K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given string **str** of length **N**. The task is to find the minimum characters to be added at the front to make string palindrome.

Note: A palindrome is a word which reads the same backward as forward.

Example: "madam".

Example 1:

Input:

S = "abc"

Output: 2

Explanation:

Add 'b' and 'c' at front of above string to make it
palindrome : "cbabc"

Example 2:

Input:

S = "aacecaaa"

Output: 1

Explanation: Add 'a' at front of above string
to make it palindrome : "aaacecaaa"

Your Task:

You don't need to read input or print anything. Your task is to complete the function `minChar()` which takes a string S and returns an integer as output.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq S.length \leq 10^6$

Expression Add Operators

Hard Accuracy: **27.11%** Submissions: **107+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string S that contains only digits and an integer **target**, return **all possibilities** strings to insert the binary operator '+', '-', and/or '*' between the digits of S so that the resultant expression evaluates to the **target** value.

Note : Operands in the returned expressions **should not** contain leading zeros.

Example 1:

Input:

$S = "123"$

$target = 6$

Output: { "1*2*3", "1+2+3" }

Explanation: Both "1*2*3" and "1+2+3" evaluate to 6.

Example 2:

Input:

$S = "232"$

$target = 8$

Output: { "2*3+2", "2+2*3" }

Explanation: Both "2*3+3" and "2+2*3" evaluate to 8.

Example 3:

Input:

$S = "3456237490"$

$target = 9191$

Output: { }

Explanation: There are no expressions that can be created from "3456237490" to evaluate to 9191.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `addOperators()` which takes string `S` and integer `target` as parameters and return a list of strings which contains all valid possibilities.

Expected Time Complexity: $O(|S| \cdot 4^{|S|})$

Expected Auxiliary Space: $O(|S| \cdot 3^{|S|})$

Constraints:

$$1 \leq |S| \leq 10$$

`S` consists of only digits.

$$-2^{31} \leq \text{target} \leq 2^{31}-1$$

Better String

Hard Accuracy: **12.56%** Submissions: **224+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a pair of strings, Geek wants to find the better string. The better string is the string having more number of **distinct** subsequences.

If both the strings have equal count of distinct subsequence then return `str1`.

Example 1:

Input:

```
str1 = "gfg", str2 = "ggg"
```

Output: "gfg"

Explanation: "gfg" have 7 distinct subsequences whereas "ggg" have 4 distinct subsequences.

Example 2:

Input: str1 = "a", str2 = "b"

Output: "a"

Explanation: Both the strings have only 1 distinct subsequence.

Constraints:

$1 \leq \text{str1.length}, \text{str2.length} \leq 30$

Your Task:

You don't need to read input or print anything. Your task is to complete the function **betterString()** which takes **str1** and **str2** as input parameters and returns the better string.

Expected Time Complexity: $O(n)$

Expected Auxiliary Space: $O(n)$

Count the Substring

Hard Accuracy: **0.0%** Submissions: **0** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a binary string **S** consists only of 0s and 1s. The task is to calculate the number of substrings that have more 1s than 0s.

Example 1:

Input:

S = "011"

Output: 4

Explanation: There are 4 substring which has more 1s than 0s. i.e "011", "1", "11" and "1"

Example 2:

Input:

S = "0000"

Output: 0

Explanation: There is no substring which has more 1s than 0s

Your Task:

You dont need to read input or print anything. Complete the function **countSubstring()** which takes the string S as input parameter and returns the number of substring which has more 1s than 0s.

Expected Time Complexity: $O(|S|)$

Expected Auxiliary Space: $O(|S|)$

Constraints:

$1 \leq |S| \leq 10^5$

|S| denotes the length of the string S

Next Happy Number

Hard Accuracy: **51.01%** Submissions: **6K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

For a given non-negative integer N , find the next smallest Happy Number. A number is called happy if it leads to 1 after a sequence of steps wherein each step number is replaced by the sum of squares of its digit that is if we start with Happy Number and keep replacing it with digits square sum, we reach 1.

Example 1:**Input:**

$N = 8$

Output:

10

Explanation:

Next happy number
after 8 is 10 because

$$1*1 + 0*0 = 1$$

Example 2:**Input:**

$N = 10$

Output

13

Explanation:

after 10, 13 is a happy
number because

$$1*1 + 3*3 = 10$$

$$1*1 + 0*0 = 1$$

Your Task:

You don't need to read input or print anything. Your task is to complete the

function **nextHappy()** which takes an integer **N** as input parameters and returns an integer, next Happy number after **N**.

Expected Time Complexity: $O(N \log_{10} N)$

Expected Space Complexity: $O(1)$

Constraints:

$1 \leq N \leq 10^3$

Check Tree Traversal

Hard Accuracy: **28.09%** Submissions: **9K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given Preorder, Inorder and Postorder traversals of some tree of size **N**. The task is to check if they are all of the same tree or not.

Example 1:

Input:

$N = 5$

$\text{preorder}[] = \{1, 2, 4, 5, 3\}$

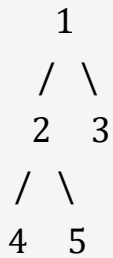
$\text{inorder}[] = \{4, 2, 5, 1, 3\}$

$\text{postorder}[] = \{4, 5, 2, 3, 1\}$

Output: Yes

Explanation:

All of the above three traversals are of the same tree.

**Example 2:****Input:**

N = 5

preorder[] = {1, 5, 4, 2, 3}

inorder[] = {4, 2, 5, 1, 3}

postorder[] = {4, 1, 2, 3, 5}

Output: No

Explanation: The three traversals can not be of the same tree.

Your Task:

You don't need to read input or print anything. Complete the function **checktree()** which takes the array **preorder[]**, **inorder[]**, **postorder[]** and integer **N** as input parameters and returns true if the three traversals are of the same tree or not.

Expected Time Complexity: $O(N^2)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 10^3$

Node values are unique.

Word Break - Part 2

Hard Accuracy: **57.49%** Submissions: **26K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string *s* and a dictionary of words *dict* of length *n*, add spaces in *s* to construct a sentence where each word is a valid dictionary word. Each dictionary word can be used more than once. Return all such possible sentences.

Follow examples for better understanding.

Example 1:

Input: *s* = "catsanddog", *n* = 5

dict = {"cats", "cat", "and", "sand", "dog"}

Output: (cats and dog)(cat sand dog)

Explanation: All the words in the given sentences are present in the dictionary.

Example 2:

Input: *s* = "catsanddog", *n* = 5

dict = {"cats", "cat", "and", "sand", "dog"}

Output: Empty

Explanation: There is no possible breaking of the string *s* where all the words are present in *dict*.

Your Task:

You do not need to read input or print anything. Your task is to complete the function **wordBreak()** which takes *n*, *dict* and *s* as input parameters and returns a list of possible sentences. If no sentence is possible it returns an empty list.

Expected Time Complexity: $O(N^2 \cdot n)$ where $N = |s|$

Expected Auxiliary Space: $O(N^2)$

Constraints:

$1 \leq n \leq 20$

$1 \leq \text{dict}[i] \leq 15$

$1 \leq |s| \leq 500$

Partition array to K subsets

Hard Accuracy: **32.52%** Submissions: **30K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an integer array $a[]$ of N elements and an integer K , the task is to check if the array $a[]$ could be divided into K non-empty subsets with equal sum of elements.

Note: All elements of this array should be part of exactly one partition.

Example 1:

Input:

$N = 5$

$a[] = \{2, 1, 4, 5, 6\}$

$K = 3$

Output:

1

Explanation: we can divide above array into 3 parts with equal sum as (2, 4), (1, 5), (6)

Example 2:

Input:

$N = 5$

$a[] = \{2, 1, 5, 5, 6\}$

$K = 3$

Output:

0

Explanation: It is not possible to divide above array into 3 parts with equal sum.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `isKPartitionPossible()` which takes the array `a[]`, the size of the array `N`, and the value of `K` as inputs and returns `true`(same as 1) if possible, otherwise `false`(same as 0).

Expected Time Complexity: $O(N \cdot 2^N)$.

Expected Auxiliary Space: $O(2^N)$.

Constraints:

$1 \leq K \leq N \leq 10$

$1 \leq a[i] \leq 100$

Print all LCS sequences

Hard Accuracy: **19.93%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given two strings `s` and `t`. Now your task is to print all longest common sub-sequences in lexicographical order.

Example 1:

Input: s = abaaa, t = baabaca

Output: aaaa abaa baaa

Example 2:

Input: s = aaa, t = a

Output: a

Your Task:

You do not need to read or print anything. Your task is to complete the function `all_longest_common_subsequences()` which takes string a and b as first and second parameter respectively and returns a list of strings which contains all possible longest common subsequences in lexicographical order.

Expected Time Complexity: $O(n^4)$

Expected Space Complexity: $O(K * n)$ where K is a constant less than n.

Constraints:

$1 \leq \text{Length of both strings} \leq 50$

N-Queen Problem

Hard Accuracy: **35.43%** Submissions: **58K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

The n-queens puzzle is the problem of placing **n** queens on a (**n×n**) chessboard such that no two queens can attack each other.

Given an integer n, find all distinct solutions to the n-queens puzzle. Each solution contains distinct board configurations of the n-queens' placement,

where the solutions are a permutation of $[1, 2, 3, \dots, n]$ in increasing order, here the number in the i th place denotes that the i th-column queen is placed in the row with that number. For eg below figure represents a chessboard $[3 \ 1 \ 4 \ 2]$.

Example 1:

Input:

1

Output:

[1]

Explanation:

Only one queen can be placed in the single cell available.

Example 2:

Input:

4

Output:

[2 4 1 3] [3 1 4 2]

Explanation:

These are the 2 possible solutions.

Your Task:

You do not need to read input or print anything. Your task is to complete the function **nQueen()** which takes n as input parameter and returns a list containing

all the possible chessboard configurations in sorted order. Return an empty list if no solution exists.

Expected Time Complexity: $O(n!)$

Expected Auxiliary Space: $O(n^2)$

Constraints:

$1 \leq n \leq 10$

Bit Difference

Hard Accuracy: **20.41%** Submissions: **36K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

We define $f(X, Y)$ as number of different corresponding bits in binary representation of X and Y . For example, $f(2, 7) = 2$, since binary representation of 2 and 7 are 010 and 111, respectively. The first and the third bit differ, so $f(2, 7) = 2$.

You are given an array A of N integers, A_1, A_2, \dots, A_N . Find sum of $f(A_i, A_j)$ for all ordered pairs (i, j) such that $1 \leq i, j \leq N$. Return the answer modulo 10^9+7 .

Example 1:

Input: $N = 2$

$A = \{2, 4\}$

Output: 4

Explaintion: We return

$$\begin{aligned} &f(2, 2) + f(2, 4) + \\ &f(4, 2) + f(4, 4) = \\ &0 + 2 + \\ &2 + 0 = 4. \end{aligned}$$
Example 2:**Input:** $N = 3$ $A = \{1, 3, 5\}$ **Output:** 8**Explanation:** We return
$$\begin{aligned} &f(1, 1) + f(1, 3) + f(1, 5) + \\ &f(3, 1) + f(3, 3) + f(3, 5) + \\ &f(5, 1) + f(5, 3) + f(5, 5) = \\ &0 + 1 + 1 + \\ &1 + 0 + 2 + \\ &1 + 2 + 0 = 8. \end{aligned}$$
Your Task:

You do not need to read input or print anything. Your task is to complete the function `countBits()` which takes the value N and the array A as input parameters and returns the desired count modulo 10^9+7 .

Expected Time Complexity: $O(N * \log_2(\text{Max}(A_i)))$ **Expected Auxiliary Space:** $O(1)$ **Constraints:** $1 \leq N \leq 10^5$ $2^0 \leq A[i] < 2^{31}$ **Reorder List****Hard** Accuracy: **47.9%** Submissions: **49K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a singly linked list: $A_0 \rightarrow A_1 \rightarrow \dots \rightarrow A_{n-2} \rightarrow A_{n-1}$, reorder it to: $A_0 \rightarrow A_{n-1} \rightarrow A_1 \rightarrow A_{n-2} \rightarrow A_2 \rightarrow A_{n-3} \rightarrow \dots$

For example: Given $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ its reorder is $1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 3$.

Note: It is recommended do this in-place without altering the node's values.

Example 1:

Input:

LinkedList: $1 \rightarrow 2 \rightarrow 3$

Output: 1 3 2

Explanation:

Here $n=3$, so the correct order is $A_0 \rightarrow A_2 \rightarrow A_1$

Example 2:

Input:

Explanation: $1 \rightarrow 7 \rightarrow 3 \rightarrow 4$

Output: 1 4 7 3

Explanation:

Here $n=4$, so the correct order is $A_0 \rightarrow A_3 \rightarrow A_1 \rightarrow A_2$

Your Task:

The task is to complete the function **reorderList()** which should reorder the list as required.

The reorder list is automatically printed by the driver's code.

Note: Try to solve without using any auxilliary space.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq N \leq 5 \cdot 10^5$

$0 \leq A[i] \leq 10^5$

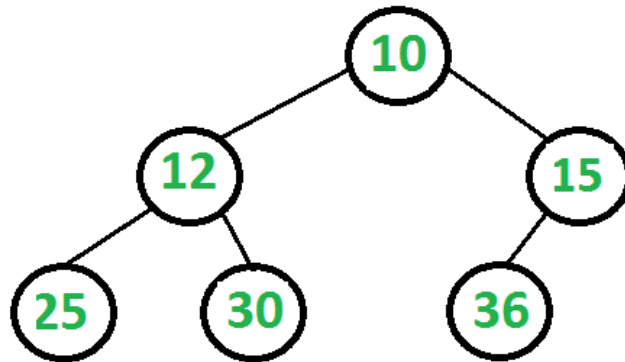
Binary Tree to DLL

Hard Accuracy: **53.36%** Submissions: **108K+** Points: **8**

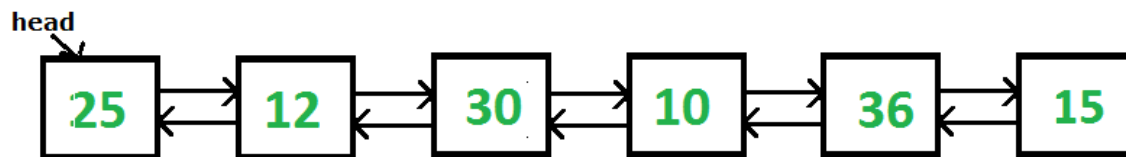
Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a Binary Tree (BT), convert it to a Doubly Linked List(DLL) In-Place. The left and right pointers in nodes are to be used as previous and next pointers respectively in converted DLL. The order of nodes in DLL must be same as Inorder of the given Binary Tree. The first node of Inorder traversal (leftmost node in BT) must be the head node of the DLL.

Note: H is the height of the tree and this space is used implicitly for the recursion stack.



The above tree should be in-place converted to following Doubly Linked List(DLL).



Example 1:

Input:

```

  1
 / \
3   2

```

Output:

```

3 1 2
2 1 3

```

Explanation: DLL would be $3 \rightleftharpoons 1 \rightleftharpoons 2$

Example 2:

Input:

```

  10
 / \
20 30
 / \

```

```
40 60
```

Output:

```
40 20 60 10 30
```

```
30 10 60 20 40
```

Explanation: DLL would be

```
40<=>20<=>60<=>10<=>30.
```

Your Task:

You don't have to take input. Complete the function **bToDLL()** that takes **root** node of the tree as a parameter and returns the head of DLL . The driver code prints the DLL both ways.

Expected Time Complexity: $O(N)$.**Expected Auxiliary Space:** $O(H)$.**Constraints:** $1 \leq \text{Number of nodes} \leq 10^5$ $0 \leq \text{Data of a node} \leq 10^5$ **Clone a linked list with next and random pointer****Hard** Accuracy: **64.8%** Submissions: **49K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given a special linked list with **N** nodes where each node has a next pointer pointing to its next node. You are also given **M** random pointers, where you will be given **M** number of pairs denoting two nodes **a** and **b** i.e. **a->arb = b**.

Construct a copy of the given list. The copy should consist of exactly **N** new nodes, where each new node has its value set to the value of its corresponding original node. Both the next and random pointer of the new nodes should point to new

nodes in the copied list such that the pointers in the original list and copied list represent the same list state. None of the pointers in the new list should point to nodes in the original list.

For example, if there are two nodes **X** and **Y** in the original list, where **X.random --> Y**, then for the corresponding two nodes **x** and **y** in the copied list, **x.random --> y**.

Return the head of the copied linked list.

Example 1:

Input:

N = 4, M = 2

value = {1,2,3,4}

pairs = {{1,2},{2,4}}

Output: 1

Explanation: In this test case, there are 4 nodes in linked list. Among these 4 nodes, 2 nodes have arbitrary pointer set, rest two nodes have arbitrary pointer as NULL. Second line tells us the value of four nodes. The third line gives the information about arbitrary pointers. The first node arbitrary pointer is set to node 2. The second node arbitrary pointer is set to node 4.

Example 2:

Input:

N = 4, M = 2


```
value[] = {1,3,5,9}
```

```
pairs[] = {{1,1},{3,4}}
```

Output: 1

Explanation: In the given testcase ,
applying the method as stated in the
above example, the output will be 1.

Your Task:

The task is to complete the function `copyList()` which takes one argument the head of the linked list to be cloned and should **return** the head of the cloned linked list.

NOTE :

1. If there is any node whose arbitrary pointer is not given then it's by default NULL.
2. Your solution return an output **1** if your clone linked list is correct, else it returns **0**.

Expected Time Complexity : $O(n)$

Expected Auxilliary Space : $O(1)$

Constraints:

$1 \leq N \leq 100$

$1 \leq M \leq N$

$1 \leq a, b \leq 100$

Subtraction in Linked List

Hard Accuracy: **12.94%** Submissions: **23K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two linked lists that represent two large positive numbers. The task is to subtract the given two numbers represented by the linked list. Subtract the smaller from the larger one.

Example 1:**Input:**

L1 = 1->0->0

L2 = 1->2

Output: 8 8

Explanation: 12 subtracted from 100 gives us 88 as result.

Your Task:

The task is to complete the function **subLinkedList()** which should subtract the numbers represented by the linked list and return the head of the linked list representing the result.

Expected Time Complexity: $O(N)$.

Expected Auxiliary Space: $O(N)$.

Constraints:

$1 \leq \text{Length of the Linked List} \leq 10000$

Sorted Link List to BST

Hard Accuracy: **53.24%** Submissions: **14K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a Singly Linked List which has data members sorted in ascending order. Construct a Balanced Binary Search Tree which has same data members as the given Linked List.

Note: There might be nodes with the same value.

Example 1:

Input:

Linked List: 1->2->3->4->5->6->7

Output:

4 2 1 3 6 5 7

Explanation :

The BST formed using elements of the linked list is,

```
      4
     / \
    2   6
   / \ / \
  1  3 5  7
```

Hence, preorder traversal of this tree is 4 2 1 3 6 5 7

Example 2:

Input:

Linked List : 1->2->3->4

Ouput:

3 2 1 4

Explanation:

The BST formed using elements of the linked list is,

```
      3
```

```
  / \
 2  4
 /
1
```

Hence, the preorder traversal of this tree is 3 2 1 4

Your task :

You don't have to read input or print anything. Your task is to complete the function **sortedListToBST()**, which takes **head** of the linked list as an input parameter and returns the root of the BST created.

Expected Time Complexity: $O(N)$, N = number of Nodes

Expected Auxiliary Space: $O(N)$, N = number of Nodes

Constraints:

$1 \leq \text{Number of Nodes} \leq 10^6$

$1 \leq \text{Value of each node} \leq 10^6$

Reverse a sublist of a linked list

Hard Accuracy: **69.78%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a linked list and positions m and n . Reverse the linked list from position m to n .

Example 1:**Input :**

N = 10

Linked List = 1->7->5->3->9->8->10
->2->2->5->NULL

m = 1, n = 8

Output : 2 10 8 9 3 5 7 1 2 5

Explanation :

The nodes from position 1 to 8
are reversed, resulting in
2 10 8 9 3 5 7 1 2 5.

Example 2:**Input:**

N = 6

Linked List = 1->2->3->4->5->6->NULL

m = 2, n = 4

Output: 1 4 3 2 5 6

Explanation:

Nodes from position 2 to 4
are reversed resulting in
1 4 3 2 5 6.

Your task :

You don't need to read input or print anything. Your task is to complete the function **reverseBetween()** which takes the head of the linked list and two integers m and n as input and returns the head of the new linked list after reversing the nodes from position m to n.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq N \leq 10^5$

Maximum of minimum for every window size

Hard Accuracy: **42.9%** Submissions: **30K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an integer array. The task is to find the maximum of the minimum of every window size in the array.

Note: Window size varies from 1 to the size of the Array.

Example 1:

Input:

$N = 7$

$arr[] = \{10, 20, 30, 50, 10, 70, 30\}$

Output: 70 30 20 10 10 10 10

Explanation:

1. First element in output

indicates maximum of minimums of all windows of size 1.

2. Minimums of windows of size 1 are {10},

{20}, {30}, {50}, {10}, {70} and {30}.

Maximum of these minimums is 70.

3. Second element in output indicates maximum of minimums of all windows of size 2.

4. Minimums of windows of size 2 are {10}, {20}, {30}, {10}, {10}, and {30}.

5. Maximum of these minimums is 30

Third element in output indicates maximum of minimums of all windows of size 3.

6. Minimums of windows of size 3 are {10}, {20}, {10}, {10} and {10}.

7. Maximum of these minimums is 20.

Similarly other elements of output are computed.

Example 2:

Input:

N = 3

arr[] = {10,20,30}

Output: 30 20 10

Explanation: First element in output indicates maximum of minimums of all windows of size 1. Minimums of windows of size 1 are {10}, {20}, {30}.

Maximum of these minimums are 30 and similarly other outputs can be computed

Your Task:

The task is to complete the function `maxOfMin()` which takes the array `arr[]` and its size `N` as inputs and finds the maximum of minimum of every window size and returns an array containing the result.

Expected Time Complexity : $O(N)$

Expected Auxilliary Space : $O(N)$

Constraints:

$1 \leq N \leq 10^5$

$1 \leq \text{arr}[i] \leq 10^6$

Secret Cipher

Hard Accuracy: **31.56%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Geek wants to send an encrypted message in the form of string `S` to his friend Keeg along with instructions on how to decipher the message. To decipher the message, his friend needs to iterate over the message string from left to right, if he finds a '*', he must remove it and add all the letters read so far to the string. He must keep on doing this till he gets rid of all the '*'.

Can you help Geek encrypt his message string `S`?

Note: If the string can be encrypted in multiple ways, find the smallest encrypted string.

Example 1:

Input: `S = "ababcbababcd"`

Output: `ab*c*d`

Explanation: We can encrypt the string in following way : "ababcbababcd" -> "ababc*d" -> "ab*c*d"

Example 2:

Input: S = "zzzzzzzz"

Output: z*z*z

Explanation: The string can be encrypted in 2 ways: "z*z*z" and "z**zzz". Out of the two "z*z*z" is smaller in length.

Your Task:

You don't need to read input or print anything. Complete the function `secretCipher()` which takes the message string S as input parameter and returns the shortest possible encrypted string.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq |S| \leq 10^5$

132 Geeky Buildings

Hard Accuracy: **21.32%** Submissions: **16K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

There are **N** buildings in Linear Land. They appear in a linear line one after the other and their heights are given in the array `arr[]`. Geek wants to select **three** buildings in Linear Land and remodel them as recreational spots. The

third of the selected building must be taller than the first and shorter than the second.

Can geek build the three-building recreational zone?

Example 1:

Input:

N = 6

arr[] = {4, 7, 11, 5, 13, 2}

Output:

True

Explanation:

[4, 7, 5] fits the condition.

Example 2:

Input:

N = 4

arr[] = {11, 11, 12, 9}

Output:

False

Explanation:

No 3 buildings fit the given condition.

Your Task:

You don't need to read input or print anything. Complete the function **recreationalSpot()** which takes the array arr[] and its size N as input parameters and returns a boolean value based on whether his building selection was successful or not.

Note: The generated output will be "True" or "False".

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 10^4$

$1 \leq \text{arr}[i] \leq 10^5$

IPL 2021 - Match Day 6 - Semi Final

Hard Accuracy: **18.16%** Submissions: **579+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

IPL 2021 knockouts are over, teams MI, CSK, DC, and RCB are qualified for the semis.

Today is matchday 6 and it is between Delhi Capitals and Royal Challengers Bangalore. Glenn Maxwell of RCB playing flawlessly. Rishabh Pant, the new captain of the team who is also a wicket-keeper wants to send a message to the bowler. But, he can't shout message directly as a batsman can hear. So, he decided to encrypt the message by putting '*'s in the message. And this is how the bowler decrypts the message. Bowler iterates over the message string from left to right, if he finds a '*', he removes it and adds all the letters read so far to the message. He keeps on doing this till he gets rid of all the '*'. Given a decrypted message in the form of the string, the task is to find the encrypted message.

Note: If the string can be encrypted in multiple ways, find the encrypted string of smallest length.

Example 1:

Input: s = "ababcababcd"

Output: ab*c*d

Explanation: We can encrypt the string in following way : "ababcababcd" -> "ababc*d" -> "ab*c*d"

Example 2:

Input: s = "zzzzzzzz"

Output: z*z*z

Explanation: The string can be encrypted in 2 ways: "z*z*z" and "z**zzz". Out of the two "z*z*z" is smaller in length.

Your Task:

You don't need to read input or print anything. Complete the function **compress()** which takes the message string **s** as input parameter and returns the shortest possible encrypted string.

Constraints:

$1 \leq |s| \leq 10^5$

Find median in a stream

Hard Accuracy: **30.34%** Submissions: **100K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an input stream of **N** integers. The task is to insert these numbers into a new stream and find the median of the stream formed by each insertion of **x** to the new stream.

Example 1:

Input:

N = 4

X[] = 5,15,1,3

Output:

5

10

5

4

Explanation:Flow in stream : 5, 15, 1, 3

5 goes to stream --> median 5 (5)

15 goes to stream --> median 10 (5,15)

1 goes to stream --> median 5 (5,15,1)

3 goes to stream --> median 4 (5,15,1 3)

Example 2:**Input:**

N = 3

X[] = 5,10,15

Output:

5

7.5

10

Explanation:Flow in stream : 5, 10, 15

5 goes to stream --> median 5 (5)

10 goes to stream --> median 7.5 (5,10)

15 goes to stream --> median 10 (5,10,15)

Your Task:

You are required to complete the class Solution.

It should have 2 data members to represent 2 heaps.

It should have the following member functions:

1. **insertHeap()** which takes **x** as input and inserts it into the heap, the function should then call **balanceHeaps()** to balance the new heap.
2. **balanceHeaps()** does not take any arguments. It is supposed to balance the two heaps.
3. **getMedian()** does not take any arguments. It should return the current median of the stream.

Expected Time Complexity : $O(n \log n)$

Expected Auxilliary Space : $O(n)$

Constraints:

$1 \leq N \leq 10^6$

$1 \leq x \leq 10^6$

AVL Tree Insertion

Hard Accuracy: **34.35%** Submissions: **12K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a AVL tree and N values to be inserted in the tree. Write a function to insert a given value into the tree.

Example 1:

N = 3

Values to be inserted = {5,1,4}

Input:

Value to be inserted = 5

Output:

5

Input :

Value to be inserted = 1

Output:

5

/

1

Input :

Value to be inserted = 4

Output:

5

4

/ LR rotation / \

1 -----> 1 5

\

4

Your Task:

You dont need to read input or print anything. Complete the function **insertToAVL()** which takes the root of the tree and the value of the node to be inserted as input parameters and returns the root of the modified tree.

Note:

The tree will be checked after each insertion.

If it violates the properties of balanced BST, an error message will be printed

followed by the inorder traversal of the tree at that moment.
If instead all insertions are successful, inorder traversal of tree will be printed.

Expected Time Complexity: $O(\log N)$

Expected Auxiliary Space: $O(\text{height of tree})$

Constraints:

$1 \leq N \leq 500$

Merge two BST 's

Hard Accuracy: **64.95%** Submissions: **37K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two BSTs, return elements of both BSTs in **sorted** form.

Example 1:

Input:

BST1:

```
  5
 / \
3   6
/\
2   4
```

BST2:

```
  2
```



```

    /  \
   1    3
    \
     7
    /
   6

```

Output: 1 2 2 3 3 4 5 6 6 7

Explanation:

After merging and sorting the two BST we get 1 2 2 3 3 4 5 6 6 7.

Example 2:

Input:

BST1:

```

    12
   /
  9
 / \
6  11

```

BST2:

```

    8
   / \
  5  10
 /
2

```

Output: 2 5 6 8 9 10 11 12

Explanation:

After merging and sorting the two BST we get 2 5 6 8 9 10 11 12.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **merge()** which takes roots of both the BSTs as its input and returns an array of integers denoting the node values of both the BSTs in a sorted order.

Expected Time Complexity: $O(M+N)$ where M and N are the sizes of the two BSTs.

Expected Auxiliary Space: $O(\text{Height of BST1} + \text{Height of BST2})$.

Constraints:

$1 \leq \text{Number of Nodes} \leq 10^5$

AVL Tree Deletion

Hard Accuracy: **30.23%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a AVL tree and N values to be deleted from the tree. Write a function to delete a given value from the tree.

Example 1:

Tree =

```
    4
   / \
  2   6
 / \  / \
1  3 5  7
```

$N = 4$

Values to be deleted = {4,1,3,6}

Input: Value to be deleted = 4

Output:

```
  5
 /  \
2    6
/\    \
1 3    7
```

Input: Value to be deleted = 1

Output:

```
  5
 /  \
2    6
 \    \
 3    7
```

Input: Value to be deleted = 3

Output:

```
  5
 /  \
2    6
     \
      7
```

Input: Value to be deleted = 6

Output:

```
  5
```

```
 /  \  
2   7
```

Your Task:

You don't need to read input or print anything. Complete the function `deleteNode()` which takes the root of the tree and the value of the node to be deleted as input parameters and returns the root of the modified tree.

Note: The tree will be checked after each deletion.

If it violates the properties of balanced BST, an error message will be printed followed by the inorder traversal of the tree at that moment.

If instead all deletions are successful, inorder traversal of tree will be printed.

If every single node is deleted from tree, 'null' will be printed.

Expected Time Complexity: $O(\text{height of tree})$

Expected Auxiliary Space: $O(\text{height of tree})$

Constraints:

$1 \leq N \leq 500$

Merge two BST 's

Hard Accuracy: **64.95%** Submissions: **37K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two BSTs, return elements of both BSTs in **sorted** form.

Example 1:**Input:****BST1:**

```
  5
 /  \
3    6
/\
2  4
```

BST2:

```
  2
 /  \
1    3
     \
      7
     /
    6
```

Output: 1 2 2 3 3 4 5 6 6 7**Explanation:**

After merging and sorting the two BST we get 1 2 2 3 3 4 5 6 6 7.

Example 2:**Input:****BST1:**

```
  12
 /
 9
 /\
6 11
```

BST2:

```
  8
 / \
5   10
/
2
```

Output: 2 5 6 8 9 10 11 12**Explanation:**

After merging and sorting the two BST we get 2 5 6 8 9 10 11 12.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **merge()** which takes roots of both the BSTs as its input and returns an array of integers denoting the node values of both the BSTs in a sorted order.

Expected Time Complexity: $O(M+N)$ where M and N are the sizes of the two BSTs.

Expected Auxiliary Space: $O(\text{Height of BST1} + \text{Height of BST2})$.

Constraints: $1 \leq \text{Number of Nodes} \leq 10^5$ **AVL Tree Deletion****Hard** Accuracy: **30.23%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a AVL tree and N values to be deleted from the tree. Write a function to delete a given value from the tree.

Example 1:

Tree =

```
    4
   / \
  2   6
 / \  / \
1  3 5  7
```

N = 4

Values to be deleted = {4,1,3,6}

Input: Value to be deleted = 4

Output:

```
    5
   / \
  2   6
 / \   \
1  3   7
```

Input: Value to be deleted = 1

Output:

```
    5
   / \
  2   6
 \   \
  3   7
```

Input: Value to be deleted = 3

Output:

```
  5
 /  \
2    6
     \
      7
```

Input: Value to be deleted = 6

Output:

```
  5
 /  \
2    7
```

Your Task:

You don't need to read input or print anything. Complete the function `deleteNode()` which takes the root of the tree and the value of the node to be deleted as input parameters and returns the root of the modified tree.

Note: The tree will be checked after each deletion.

If it violates the properties of balanced BST, an error message will be printed followed by the inorder traversal of the tree at that moment.

If instead all deletions are successful, inorder traversal of tree will be printed.

If every single node is deleted from tree, 'null' will be printed.

Expected Time Complexity: $O(\text{height of tree})$

Expected Auxiliary Space: $O(\text{height of tree})$

Constraints:

$1 \leq N \leq 500$

Nodes at given distance in binary tree

Hard Accuracy: **32.36%** Submissions: **33K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a binary tree, a target node in the binary tree, and an integer value k, find all the nodes that are at distance k from the given target node. No parent pointers are available.

Note: You have to return list in sorted order.

Example 1:

Input:

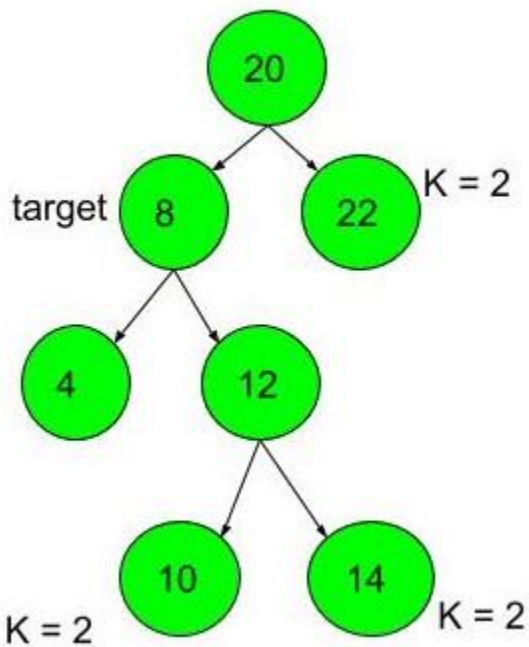
```
      20
     /  \
    8    22
   /  \
  4   12
 /  \
10  14
```

Target Node = 8

K = 2

Output: 10 14 22

Explanation: The three nodes at distance 2 from node 8 are 10, 14, 22.

**Example 2:****Input:**

```

    20
   /  \
  7    24
 /  \
4    3
 /
1

```

Target Node = 7

K = 2

Output: 1 24**Your Task:**

You don't need to read input or print anything. Complete the function **KDistanceNodes()** which takes the root of the tree, target, and K as input parameters and returns a list of nodes at k distance from target in a sorted order.

Expected Time Complexity: $O(N \cdot \log N)$

Expected Auxiliary Space: $O(\text{Height of tree})$

Constraints:

$1 \leq N \leq 10^3$

$1 \leq \text{data of node} \leq 10^4$

$1 \leq \text{target} \leq 10^4$

$1 \leq k \leq 20$

Maximum Path Sum between 2 Leaf Nodes

Hard Accuracy: **18.39%** Submissions: **157K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a binary tree in which each node element contains a number. Find the maximum possible path sum from one leaf node to another leaf node.

Note: Here Leaf node is a node which is connected to exactly one different node.

Example 1:

Input:

```
      3
     / \
    4   5
   / \
  -10 4
```

Output: 16

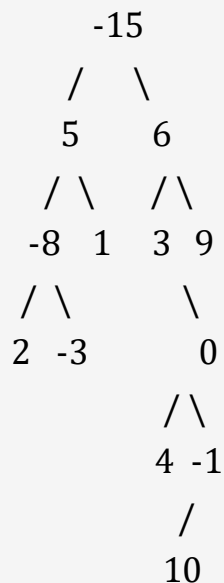
Explanation:

Maximum Sum lies between leaf node 4 and 5.

$4 + 4 + 3 + 5 = 16.$

Example 2:

Input:



Output: 27

Explanation:

The maximum possible sum from one leaf node to another is $(3 + 6 + 9 + 0 + -1 + 10 = 27)$

Your Task:

You don't need to read input or print anything. Complete the function **maxPathSum()** which takes root node as input parameter and returns the maximum sum between 2 leaf nodes.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(\text{Height of Tree})$

Constraints: $2 \leq \text{Number of nodes} \leq 10^4$ $-10^3 \leq \text{Value of each node} \leq 10^3$ **Number of Turns in Binary Tree****Hard** Accuracy: **42.63%** Submissions: **17K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a binary tree and data value of two of its nodes. Find the number of turns needed to reach from one node to another in the given binary tree.

Example 1:**Input:**

Tree =

```
      1
     / \
    2   3
   / \ / \
  4  5 6  7
 /  / \
8  9 10
```

first node = 5

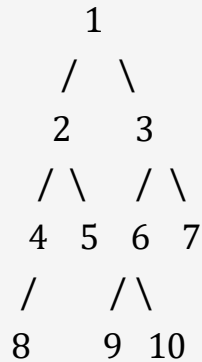
second node = 10

Output: 4**Explanation:**

Turns will be at 2, 1, 3, 6.

Example 2:**Input:**

Tree =



first node = 1

second node = 4

Output: -1**Explanation:** No turn is required since they are in a straight line.**Your Task:**

You don't need to read input or print anything. Complete the function **NumberOfTurns()** which takes root node and data value of 2 nodes as input parameters and returns the number of turns required to navigate between them. If the two nodes are in a straight line, ie- the path does not involve any turns, return -1.

Expected Time Complexity: $O(N)$ **Expected Auxiliary Space:** $O(\text{Height of Tree})$ **Constraints:** $1 \leq N \leq 10^3$ **Burning Tree**

Hard Accuracy: **53.53%** Submissions: **25K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a binary tree and a node called target. Find the minimum time required to burn the complete binary tree if the target is set on fire. It is known that in 1 second all nodes connected to a given node get burned. That is its left child, right child, and parent.

Example 1:

Input:



Target Node = 8

Output: 7

Explanation: If leaf with the value 8 is set on fire.

After 1 sec: 5 is set on fire.

After 2 sec: 2, 7 are set to fire.

After 3 sec: 4, 1 are set to fire.
After 4 sec: 3 is set to fire.
After 5 sec: 6 is set to fire.
After 6 sec: 9 is set to fire.
After 7 sec: 10 is set to fire.
It takes 7s to burn the complete tree.

Example 2:**Input:**

```
      1
     / \
    2   3
   / \  \
  4  5   7
 /  /
8  10
```

Target Node = 10
Output: 5

Your Task:

You don't need to read input or print anything. Complete the function `minTime()` which takes the root of the tree and target as input parameters and returns the minimum time required to burn the complete binary tree if the target is set on fire at the 0th second.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(\text{height of tree})$

Constraints:

$$1 \leq N \leq 10^4$$

Possible Paths

Hard Accuracy: **49.54%** Submissions: **341+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a tree with N nodes and $(N-1)$ edges such that each edge has some weight. You are given Q queries. Each query contains a number X . Find the number of paths in which the edge that has the maximum weight is less than or equals to X .

Note: Path from A to B and B to A are considered to be the same.

Example 1:

Input: $N = 7$

Tree =

```

      1
     /
    /
   /
  2
 / \
1 /  \ 9
 /  \
3    4
 / \

```

```

      7 / \ 8
      /  \
     6    5
      /
     4 /
      /
     7

```

list of edges {start,end,weight}=

{{1, 2, 3}, {2, 3, 1}, {2, 4, 9},

{3, 6, 7}, {3, 5, 8}, {5, 7, 4}}

Q = 3

queries[] = {1, 3, 5}

Output: 1 3 4

Explanation:

Query 1: Path from 2 to 3

Query 2: Path from 1 to 2, 1 to 3, and
2 to 3

Query 3: Path from 1 to 2, 1 to 3, 2 to 3,
and 5 to 7

Example 2:

Input: N = 3

list of edges {start, end, weight}=

{{1, 2, 1}, {2, 3, 4}}

Q = 1

queries[] = {3}

Output: 1

Explanation:

Query 1: Path from 1 to 2

Your Task:

You don't need to read input or print anything. Complete the function **maximumWeight()** which takes integers **N**, list of **edges** where each edge is given by {start,end,weight}, an integer **Q** and a list of **Q queries** as input parameters and returns a list of integers denoting the maximum number of possible paths for each query.

Expected Time Complexity: $O(N \log N + Q \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$$2 \leq N \leq 10^4$$

$$\text{Number of edges} = N - 1$$

$$1 \leq \text{val}[i] \leq 10^5$$

Corona Vaccine

Hard Accuracy: **60.22%** Submissions: **6K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Geek has developed an effective vaccine for Corona virus and he wants each of the **N** houses in Geek Land to have access to it. Given a binary tree where each node represents a house in Geek Land, find the minimum number of houses that should be supplied with the vaccine kit if one vaccine kit is sufficient for that house, its parent house and its immediate child nodes.

Example 1:**Input:**

```
1
 /\
2 3
  \
  4
   \
   5
    \
    6
```

Output:

2

Explanation:

The vaccine kits should be supplied to house numbers 1 and 5.

Example 2:**Input:**

```
1
 /\
2 3
```

Output:

1

Explanation:

The vaccine kits should be supplied to house number 1.

Your Task:

You don't need to read input or print anything. Complete the function **supplyVaccine()** which takes the root of the housing tree as input parameter and returns the minimum number of houses that should be supplied with the vaccine kit.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$$1 \leq N \leq 10^5$$

Shortest Range In BST

Hard Accuracy: **62.56%** Submissions: **2K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a BST (Binary Search Tree), find the shortest range $[x, y]$, such that, at least one node of every level of the BST lies in the range.

If there are multiple ranges with the same gap (i.e. $(y-x)$) return the range with the **smallest** x .

Example 1:

Input:

8
/ \

```

    3  10
   / \  \
  2  6  14
   /\  /
  4 7 12
     /\
    11 13

```

Output: 6 11

Explanation: Level order traversal of the tree is [8], [3, 10], [2, 6, 14], [4, 7, 12], [11, 13].

The shortest range which satisfies the above mentioned condition is [6, 11].

Example 2:

Input:

```

12
 \
 13
  \
  14
   \
   15
    \
    16

```

Output: 12 16

Explanation: Each level contains one node, so the shortest range is [12, 16].

Your Task:

You don't need to read input or print anything. Complete the function **shortestRange()** which takes the root of the tree as an input parameter and returns the pair of numbers

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$$1 \leq N \leq 10^5$$

$$1 \leq \text{Node Value} \leq 10^5$$

Distribute candies in a binary tree

Hard Accuracy: **59.19%** Submissions: **1K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a binary tree with N nodes, in which each node value represents number of candies present at that node. In one move, one may choose two adjacent nodes and move one candy from one node to another (the move may be from parent to child, or from child to parent.)

The task is to find the number of moves required such that every node have **exactly one** candy.

Example 1:

Input : 3
 / \

```

    0  0

```

Output : 2

Explanation:

From the root of the tree, we move one candy to its left child, and one candy to its right child.

Example 2:

Input : 0

```

    /  \
   3    0

```

Output : 3

Explanation :

From the left child of the root, we move two candies to the root [taking two moves]. Then, we move one candy from the root of the tree to the right child.

Your task :

You don't have to read input or print anything. Your task is to complete the function **distributeCandy()** which takes the root of the tree as input and returns the number of moves required such that every node has exactly one candy.

Expected Time Complexity: $O(n)$

Expected Auxiliary Space: $O(h)$

Constraints:

$1 \leq n \leq 10^4$

Check if all levels of two trees are anagrams or not

Hard Accuracy: **56.88%** Submissions: **15K+** Points: **8**

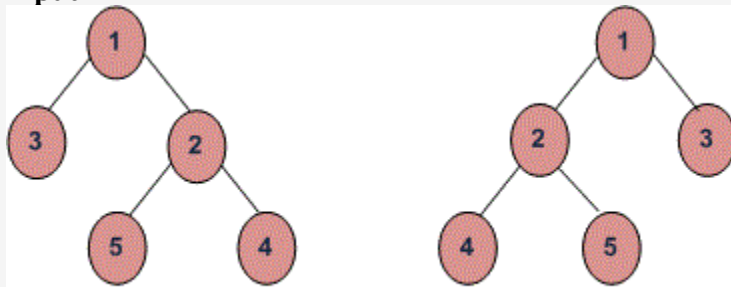
Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two binary trees with same number of nodes, the task is to check if each of their levels are anagrams of each other or not.

Note: All nodes of a tree should be unique.

Example 1:

Input:



Output: 1

Explanation:

Tree 1:

Level 0 : 1

Level 1 : 3, 2

Level 2 : 5, 4

Tree 2:

Level 0 : 1

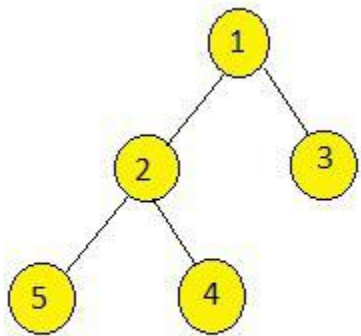
Level 1 : 2, 3

Level 2 : 4, 5

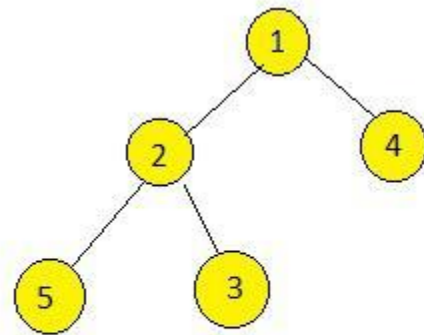
As we can clearly see all the levels of above two binary trees are anagrams of each other, hence return true.

Example 2:

Input:



Tree-1



Tree-2

Output: 0

Explanation:

Tree 1:

Level 0 : 1

Level 1 : 2, 3

Level 2 : 5, 4

Tree 2:

Level 0 : 1

Level 1 : 2, 4

Level 2 : 5, 3

As we can clearly see that level 1 and level 2 are not anagrams of each other, hence return false.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `areAnagrams()` which takes the root of two trees as input and returns an 1 if all the levels are anagrams, else returns 0 as output.

Expected Time Complexity: $O(N \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 10^4$

$1 \leq \text{tree.val} \leq 10^9$

Fixing Two nodes of a BST

Hard Accuracy: **53.68%** Submissions: **34K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given the **root** of a binary search tree(BST), where exactly two nodes were swapped by mistake. Fix (or correct) the BST by swapping them back. Do not change the structure of the tree.

Note: It is guaranteed that the given input will form BST, except for 2 nodes that will be wrong. All changes must be reflected in the original linked list.

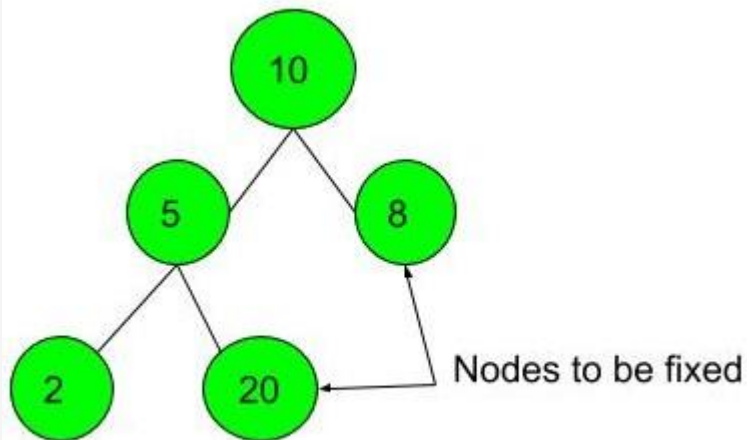
Example 1:

Input:

```
10
 / \
5   8
 /\
2  20
```

Output: 1

Explanation:



Example 2:

Input:

```
11
 / \
3   17
 \ /
 4 10
```

Output: 1

Explanation:

By swapping nodes 11 and 10, the BST can be fixed.

Your Task:

You don't need to take any input. Just complete the function **correctBst()** that takes root node as **parameter**. The function should not return anything, all the changes must be done in the existing tree only. BST will then be checked by driver code and 0 or 1 will be printed.

Expected Time Complexity: $O(n)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq \text{Number of nodes} \leq 10^3$

Optimal binary search tree

Hard Accuracy: **54.88%** Submissions: **7K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a sorted array **keys[0.. n-1]** of search keys and an array **freq[0.. n-1]** of frequency counts, where **freq[i]** is the number of searches to **keys[i]**. Construct a binary search tree of all keys such that the total cost of all the searches is as small as possible.

Let us first define the cost of a BST. The cost of a BST node is level of that node multiplied by its frequency. Level of root is 1.

Example 1:

Input:

n = 2

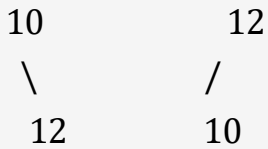
keys = {10, 12}

freq = {34, 50}

Output: 118

Explanation:

There can be following two possible BSTs



*The cost of tree I is $34*1 + 50*2 = 134$*

*The cost of tree II is $50*1 + 34*2 = 118$*

Example 2:

Input:

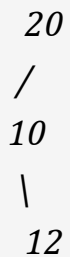
$N = 3$

keys = {10, 12, 20}

freq = {34, 8, 50}

Output: 142

Explanation: *There can be many possible BSTs*



Among all possible BSTs,

cost of this BST is minimum.

*Cost of this BST is $1*50 + 2*34 + 3*8 = 142$*

Your Task:

You don't need to read input or print anything. Your task is to complete the function `optimalSearchTree()` which takes the array `keys[]`, `freq[]` and their size `n` as input parameters and returns the total cost of all the searches is as small as possible.

Expected Time Complexity: $O(n^3)$

Expected Auxiliary Space: $O(n^2)$

Constraints:

$1 \leq N \leq 100$

Minimum Cost Path

Hard Accuracy: **26.99%** Submissions: **57K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a square **grid** of size **N**, each cell of which contains integer cost which represents a cost to traverse through that cell, we need to find a path from top left cell to bottom right cell by which the total cost incurred is minimum.

From the cell (i, j) we can go $(i, j-1)$, $(i, j+1)$, $(i-1, j)$, $(i+1, j)$.

Note: It is assumed that negative cost cycles do not exist in the input matrix.

Example 1:

Input: grid = {{9,4,9,9},{6,7,6,4},

$\{8,3,3,7\},\{7,4,9,10\}$

Output: 43

Explanation: The grid is-

9 4 9 9

6 7 6 4

8 3 3 7

7 4 9 10

The minimum cost is-

$9 + 4 + 7 + 3 + 3 + 7 + 10 = 43$.

Example 2:

Input: grid = $\{\{4,4\},\{3,7\}\}$

Output: 14

Explanation: The grid is-

4 4

3 7

The minimum cost is- $4 + 3 + 7 = 14$.

Your Task:

You don't need to read or print anything. Your task is to complete the function **minimumCostPath()** which takes grid as input parameter and returns the minimum cost to reach at bottom right cell from top left cell.

Expected Time Complexity: $O(n^2 \log(n))$

Expected Auxiliary Space: $O(n^2)$

Constraints:

$$1 \leq n \leq 500$$

$$1 \leq \text{cost of cells} \leq 1000$$

Minimum number of elements**Hard** Accuracy: **40.79%** Submissions: **1K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array **arr** of size **N**, the task is to make strictly increasing and strictly decreasing subsequences from the array such that each array element belongs to increasing subsequence or decreasing subsequence, but not both, or can be part of none of the subsequence. Minimize the number of elements that are not part of any of the subsequences and find the count of such elements.

Example 1:

Input: $N = 12$, $\text{arr}[] = \{7, 8, 1, 2, 4, 6, 3, 5, 2, 1, 8, 7\}$

Output: 2

Explanation:

Increasing sequence can be :

$\{1, 2, 4, 5, 8\}$.

Decreasing sequence can be :

$\{7, 6, 3, 2, 1\}$.

So, only 2 (8, 7) elements are left which are not part of either of the subsequences.

Example 2:

Input: $N = 7$, $\text{arr}[] = \{1, 4, 2, 3, 3, 2, 4\}$

Output: 0

Explanation:

Increasing sequence can be :

$\{1, 2, 3, 4\}$.

Decreasing sequence can be :

$\{4, 3, 2\}$.

Your Task:

You don't need to read input or print anything. Complete the function `minCount()` which takes N and array `arr` as input parameters and returns the integer value

Expected Time Complexity: $O(N^3)$

Expected Auxiliary Space: $O(N^3)$

Constraints:

$1 \leq N \leq 10^2$

Minimum Points To Reach Destination

Hard Accuracy: **32.48%** Submissions: **13K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a grid of size $M \times N$ with each cell consisting of an integer which represents points. We can move across a cell only if we have positive points. Whenever we pass through a cell, points in that cell are added to our overall points, the task is

to find minimum initial points to reach cell $(m-1, n-1)$ from $(0, 0)$ by following these certain set of rules :

1. From a cell (i, j) we can move to $(i + 1, j)$ or $(i, j + 1)$.
2. We cannot move from (i, j) if your overall points at (i, j) are ≤ 0 .
3. We have to reach at $(n-1, m-1)$ with minimum positive points i.e., > 0 .

Example 1:

Input: $M = 3, N = 3$

```
arr[][] = {{-2,-3,3},  
           {-5,-10,1},  
           {10,30,-5}};
```

Output: 7

Explanation: 7 is the minimum value to reach the destination with positive throughout the path. Below is the path.

$(0,0) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (1, 2) \rightarrow (2, 2)$

We start from $(0, 0)$ with 7, we reach

$(0, 1)$ with 5, $(0, 2)$ with 2, $(1, 2)$

with 5, $(2, 2)$ with and finally we have

1 point (we needed greater than 0 points at the end).

Example 2:

Input: $M = 3, N = 2$

```
arr[][] = {{2,3},  
           {5,10},  
           {10,30}};
```

Output: 1

Explanation: Take any path, all of them are positive. So, required one point at the start

Your Task:

You don't need to read input or print anything. Complete the function **minPoints()** which takes **N**, **M** and 2-d **vector** as input parameters and returns the integer value

Expected Time Complexity: $O(N*M)$

Expected Auxiliary Space: $O(N*M)$

Constraints:

$$1 \leq N \leq 10^3$$

Minimum sum partition

Hard Accuracy: **38.97%** Submissions: **109K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array **arr** of size **n** containing **non-negative** integers, the task is to divide it into two sets **S1** and **S2** such that the absolute difference between their sums is minimum and find the **minimum** difference

Example 1:

Input: $N = 4$, $\text{arr}[] = \{1, 6, 11, 5\}$

Output: 1

Explanation:

Subset1 = {1, 5, 6}, sum of Subset1 = 12

Subset2 = {11}, sum of Subset2 = 11

Example 2:

Input: N = 2, arr[] = {1, 4}

Output: 3

Explanation:

Subset1 = {1}, sum of Subset1 = 1

Subset2 = {4}, sum of Subset2 = 4

Your Task:

You don't need to read input or print anything. Complete the function **minDifference()** which takes N and array **arr** as input parameters and returns the integer value

Expected Time Complexity: $O(N * |\text{sum of array elements}|)$

Expected Auxiliary Space: $O(N * |\text{sum of array elements}|)$

Constraints:

$1 \leq N * |\text{sum of array elements}| \leq 10^6$

$0 < \text{arr}[i] \leq 10^5$

Number Formation

Hard Accuracy: **46.1%** Submissions: **16K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given three integers x , y , and z , the task is to find the sum of all the numbers formed by having 4 at most x times, having 5 at most y times, and having 6 at most z times as a digit.

Note: Output the sum modulo 10^9+7 .

Example 1:

Input: $X = 1, Y = 1, Z = 1$

Output: 3675

Explanation: $4 + 5 + 6 + 45 + 54 + 56$
 $+ 65 + 46 + 64 + 456 + 465$
 $+ 546 + 564 + 645 + 654 = 3675$

Example 2:

Input: $X = 0, Y = 0, Z = 0$

Output: 0

Explanation: No number can be formed

Your Task:

You don't need to read input or print anything. Complete the function `getSum()` which takes X , Y and Z as input parameters and returns the integer value

Expected Time Complexity: $O(X*Y*Z)$

Expected Auxiliary Space: $O(X*Y*Z)$

Constraints:

$0 \leq X, Y, Z \leq 60$

Palindromic Partitioning

Hard Accuracy: **27.82%** Submissions: **76K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a string **str**, a partitioning of the string is a palindrome partitioning if every sub-string of the partition is a palindrome. Determine the fewest cuts needed for palindrome partitioning of the given string.

Example 1:

Input: str = "ababbbabbababa"

Output: 3

Explanation: After 3 partitioning substrings are "a", "babbbab", "b", "ababa".

Example 2:

Input: str = "aaabba"

Output: 1

Explanation: The substrings after 1 partitioning are "aa" and "abba".

Your Task:

You do not need to read input or print anything, Your task is to complete the function **palindromicPartition()** which takes the string **str** as the input parameter and returns the minimum number of partitions required.

Expected Time Complexity: $O(n*n)$ [n is the length of the string str]

Expected Auxiliary Space: $O(n*n)$

Constraints:

$1 \leq \text{length of str} \leq 500$

Maximum Profit

Hard Accuracy: **48.35%** Submissions: **25K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

In the stock market, a person buys a stock and sells it on some future date. Given the stock prices of N days in an array $A[]$ and a positive integer K , find out the maximum profit a person can make in at-most K transactions. A transaction is equivalent to (buying + selling) of a stock and new transaction can start only when the previous transaction has been completed.

Example 1:

Input: $K = 2, N = 6$

$A = \{10, 22, 5, 75, 65, 80\}$

Output: 87

Explanation:

1st transaction: buy at 10 and sell at 22.

2nd transaction : buy at 5 and sell at 80.

Example 2:

Input: $K = 3, N = 4$

$A = \{20, 580, 420, 900\}$

Output: 1040

Explanation: The trader can make at most 2 transactions and giving him a profit of 1040.

Example 3:

Input: $K = 1, N = 5$

$A = \{100, 90, 80, 50, 25\}$

Output: 0

Explanation: Selling price is decreasing daily. So seller cannot have profit.

Your Task:

You do not need to read input or print anything. Your task is to complete the function **maxProfit()** which takes the values K, N and the array $A[]$ as input parameters and returns the maximum profit.

Expected Time Complexity: $O(N \cdot K)$

Expected Auxiliary Space: $O(N \cdot K)$

Constraints:

$1 \leq N \leq 500$

$1 \leq K \leq 200$

$1 \leq A[i] \leq 1000$

Binary Search

Basic Accuracy: **44.32%** Submissions: **271K+** Points: **1**

Unlock your potential! Upskill yourself with Data Structures and Algorithms

Given a sorted array of size **N** and an integer **K**, find the position at which **K** is present in the array using binary search.

Example 1:**Input:**

N = 5

arr[] = {1 2 3 4 5}

K = 4

Output: 3

Explanation: 4 appears at index 3.

Example 2:**Input:**

N = 5

arr[] = {11 22 33 44 55}

K = 445

Output: -1

Explanation: 445 is not present.

Your Task:

You don't need to read input or print anything. Complete the function **binarysearch()** which takes **arr[]**, **N** and **K** as input parameters and returns the index of **K** in the array. If **K** is not present in the array, return -1.

Expected Time Complexity: $O(\log N)$

Expected Auxiliary Space: $O(\log N)$ if solving recursively and $O(1)$ otherwise.

Constraints:

- $1 \leq N \leq 10^5$
- $1 \leq \text{arr}[i] \leq 10^6$
- $1 \leq K \leq 10^6$

Strictly Increasing Array**Hard** Accuracy: **26.64%** Submissions: **11K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array **nums[]** of **N** positive integers. Find the minimum number of operations required to modify the array such that array elements are in strictly increasing order ($A[i] < A[i+1]$).

Changing a number to greater or lesser than original number is counted as one operation.

Example 1:**Input:** `nums[] = [1, 2, 3, 6, 5, 4]`**Output:** 2

Explanation: By decreasing 6 by 2 and increasing 4 by 2, arr will be like [1, 2, 3, 4, 5, 6] which is strictly increasing.

Example 2:

Input: `nums[] = [1, 2, 3, 4]`

Output: 0

Explanation: Arrays is already strictly increasing.

Your Task:

You don't need to read or print anything. Your task is to complete the function `min_operations()` which takes the array `nums[]` as input parameter and returns the minimum number of operation needed to make the array strictly increasing.

Expected Time Complexity: $O(n^2)$

Expected Space Complexity: $O(n)$

Constraints:

$1 \leq \text{length of array} \leq 1000$

$1 \leq \text{arr}[i] \leq 1000000$

Boolean Parenthesization

Hard Accuracy: **20.15%** Submissions: **65K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a boolean expression `S` of length `N` with following symbols.

Symbols

'T' ---> true

'F' ---> false

and following operators filled between symbols

Operators

& ---> boolean AND

| ---> boolean OR

^ ---> boolean XOR

Count the number of ways we can parenthesize the expression so that the value of expression evaluates to true.

Example 1:

Input: N = 7

S = T|T&F^T

Output: 4

Explanation: The expression evaluates to true in 4 ways ((T|T)&(F^T)), (T|(T&(F^T))), (((T|T)&F)^T) and (T|((T&F)^T)).

Example 2:

Input: N = 5

S = T^F|F

Output: 2

Explanation: ((T^F)|F) and (T^(F|F)) are the only ways.

Your Task:

You do not need to read input or print anything. Your task is to complete the function `countWays()` which takes N and S as input parameters and returns number of possible ways modulo 1003.

Expected Time Complexity: $O(N^3)$

Expected Auxiliary Space: $O(N^2)$

Constraints:

$1 \leq N \leq 200$

Longest Zig-Zag Sub Sequence

Hard Accuracy: **33.72%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array **nums** of **n** positive integers. The task is to find the longest Zig-Zag subsequence problem such that all elements of this are alternating ($\text{nums}_{i-1} < \text{nums}_i > \text{nums}_{i+1}$ or $\text{nums}_{i-1} > \text{nums}_i < \text{nums}_{i+1}$).

Example 1:

Input: `nums = {1,5,4}`

Output: 3

Explanation: The entire sequence is a Zig-Zag sequence.

Example 2:

Input: `nums = {1,17,5,10,13,15,10,5,16,8}`

Output: 7

Explanation: There are several subsequences that achieve this length.
One is {1,17,10,13,10,16,8}.

Your Task:

You don't need to read or print anything. Your task is to complete the function `ZigZagMaxLength()` which takes the sequence `nums` as input parameter and returns the maximum length of alternating sequence.

Expected Time Complexity: $O(n)$

Expected Space Complexity: $O(1)$

Constraints:

$1 \leq n \leq 10^5$

Longest Path in a matrix

Hard Accuracy: **53.95%** Submissions: **9K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a $n*m$ matrix, find the maximum length path (starting from any cell) such that all cells along the path are in strictly increasing order.

We can move in 4 directions from a given cell (i, j) , i.e., we can move to $(i+1, j)$ or $(i, j+1)$ or $(i-1, j)$ or $(i, j-1)$.

Example 1:

Input: matrix = {{1,2,9},{5,3,8},{4,6,7}}

Output: 7

Explanation: The longest increasing path is {1,2,3,6,7,8,9}.

Example 2:

Input: matrix = {{3,4,5},{3,2,6},{2,2,1}}

Output: 4

Explanation: The longest increasing path is {3,4,5,6}.

Your Task:

You don't need to read or print anything. Your task is to complete the function **longestIncreasingPath()** which takes matrix as input parameter and returns the length of the longest increasing path.

Expected Time Complexity: $O(n*m)$

Expected Space Complexity: $O(n*m)$

Constraints:

$1 \leq n, m \leq 100$

$1 \leq \text{matrix}[i][j] \leq 10^4$

Word Wrap

Hard Accuracy: **29.74%** Submissions: **31K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array **nums[]** of size **n**, where **nums[i]** denotes the number of characters in one word. Let **K** be the limit on the number of characters that can be put in one line (line width). Put line breaks in the given sequence such that the lines are printed neatly.

Assume that the length of each word is smaller than the line width. When line breaks are inserted there is a possibility that extra spaces are present in each line. The extra spaces include spaces put at the end of every line **except the last one**.

You have to **minimize** the following total cost where **total cost** = Sum of cost of all lines, where cost of line is = (Number of extra spaces in the line)².

Example 1:

Input: nums = {3,2,2,5}, k = 6

Output: 10

Explanation: Given a line can have 6 characters,

Line number 1: From word no. 1 to 1

Line number 2: From word no. 2 to 3

Line number 3: From word no. 4 to 4

So total cost = $(6-3)^2 + (6-2-2-1)^2 = 3^2 + 1^2 = 10$.

As in the first line word length = 3 thus extra spaces = $6 - 3 = 3$ and in the second line there are two word of length 2 and there already 1 space between two word thus extra spaces = $6 - 2 - 2 - 1 = 1$. As mentioned in the problem description there will be no extra spaces in the last line. Placing first and second word in first line and third word on second line would take a cost of $0^2 + 4^2 = 16$ (zero spaces

on first line and $6-2 = 4$ spaces on second), which isn't the minimum possible cost.

Example 2:

Input: nums = {3,2,2}, k = 4

Output: 5

Explanation: Given a line can have 4 characters,

Line number 1: From word no. 1 to 1

Line number 2: From word no. 2 to 2

Line number 3: From word no. 3 to 3

Same explanation as above total cost

$$= (4 - 3)^2 + (4 - 2)^2 = 5.$$

Your Task:

You don't need to read or print anything. Your task is to complete the function **solveWordWrap()** which takes nums and k as input parameter and returns the minimized total cost.

Expected Time Complexity: $O(n^2)$

Expected Space Complexity: $O(n)$

Constraints:

$$1 \leq n \leq 500$$

$$1 \leq \text{nums}[i] \leq 1000$$

$$\max(\text{nums}[i]) \leq k \leq 2000$$

Longest Increasing Path in a Matrix

Hard Accuracy: **28.96%** Submissions: **3K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a matrix with n rows and m columns. Your task is to find the length of the longest increasing path in matrix, here increasing path means that the value in the specified path increases. For example if a path of length k has values $a_1, a_2, a_3, \dots, a_k$, then for every i from $[2, k]$ this condition must hold $a_i > a_{i-1}$. No cell should be revisited in the path.

From each cell, you can either move in four directions: left, right, up, or down. You are not allowed to move diagonally or move outside the boundary.

Example 1:

Input:

$n = 3, m = 3$

matrix[][] = {{1 2 3},
 {4 5 6},
 {7 8 9}}

Output:

5

Explanation:

The longest increasing path is
{1, 2, 3, 6, 9}.

Example 2:

Input:

$n = 3, m = 3$

matrix[][] = {{3 4 5},

{6 2 6},
{2 2 1}}

Output:

4

Explanation:

The longest increasing path is
{3, 4, 5, 6}.

Your Task:

You only need to implement the given function **longestIncreasingPath()** which takes the two integers **n** and **m** and the matrix **matrix** as input parameters, and returns the length of the longest increasing path in matrix.

Expected Time Complexity: $O(n*m)$

Expected Auxiliary Space: $O(n*m)$

Constraints:

$1 \leq n, m \leq 1000$

$0 \leq \text{matrix}[i] \leq 2^{30}$

Array Partition

Hard Accuracy: **57.05%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array of **N** integers, you have to find if it is possible to partition the array with following rules:

- Each element should belong to exactly one partition.
- Each partition should have atleast **K** elements.
- Absolute difference between any pair of elements in the same partition should not exceed **M**.

Example 1:**Input:**

N = 5

K = 2

M = 3

A[] = {8, 3, 9, 1, 2}

Output:

YES

Explanation:

We can partition the array into two partitions: {8, 9} and {3, 1, 2} such that all rules are satisfied.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **partitionArray()** which takes the number of elements **N**, integer **K**, integer **M** and array **A[]** as input parameters and returns true if we can partition the array such that all rules are satisfied, else returns false.

Expected Time Complexity: $O(N * \log(N))$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 2 * 10^5$

$1 \leq K \leq N$

$$1 \leq M \leq 10^9$$

$$1 \leq A[i] \leq 10^9$$

Count occurrences of a given word in a 2-d array

Hard Accuracy: **63.72%** Submissions: **13K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Find the number of occurrences of a given search word in a 2d-Array of characters where the word can go up, down, left, right, and around 90-degree bends.

Note: While making a word you can use one cell only once.

Example 1:

Input:

R = 4, C = 5

mat = {{S,N,B,S,N},

{B,A,K,E,A},

{B,K,B,B,K},

{S,E,B,S,E}}

target = SNAKES

Output:

3

Explanation:

S N B S N

B A K E A

B K B B K

S E B S E

Total occurrence of the word is 3
and denoted by color.

Example 2:**Input:**

R = 3, C = 3

mat = {{c,a,t},
 {a,t,c},
 {c,t,a}}

target = cat

Output:

5

Explanation: Same explanation
as first example.

Your task:

You don't need to read input or print anything. Your task is to complete the function `findOccurrence()` which takes the mat containing N*M 2-d array of characters and target as input parameters and returns the number of occurrences of the target.

Expected Time Complexity: $O(R \cdot C \cdot 2^{2 \cdot \text{len}})$, Where len is the length of the target String.

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq \text{len} \leq 15$

$1 \leq R, C \leq 50$

Minimum Cost to Merge Stones

Hard Accuracy: **42.85%** Submissions: **28+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

There are N piles of **stones** arranged in a row. The i^{th} pile has **stones** $[i]$ stones.
A move consists of merging exactly K **consecutive** piles into one piles, and the cost of this move is equal to the total number of stones in these K piles.
Return the **minimum** cost to merge all piles of stones into one pile. If it is impossible, return **-1** .

Example 1:

Input:

$N = 4$

$K = 2$

stones $[] = \{3, 2, 4, 1\}$

Output: 20

Explanation: We start with $\{3, 2, 4, 1\}$

We merge $\{3, 2\}$ for a cost of 5, and we are left with $\{5, 4, 1\}$.

We merge $\{4, 1\}$ for a cost of 5, and we are left with $\{5, 5\}$.

We merge $\{5, 5\}$ for a cost of 10, and we are left with $\{10\}$.

The total cost was 20, and it is proven that this is the minimum possible cost.

Example 2:

Input:

$N = 4$

$K = 3$

stones $[] = \{3, 2, 4, 1\}$

Output: -1

Explanation: After any merge operation, there are 2 piles left, and we can't merge anymore. So the task is impossible.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `mergeStones()` which takes the array of integers `stones`, integer `N` and an integer `K` as parameters and returns the minimum cost to merge all stones.

Expected Time Complexity: $O(N^3)$

Expected Auxiliary Space: $O(N^3)$

Constraints:

$$1 \leq N \leq 30$$

$$2 \leq K \leq 30$$

$$1 \leq \text{stones}_i \leq 100$$

Chocolates Pickup

Hard Accuracy: **62.28%** Submissions: **114+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given an `r` rows and `c` cols matrix `grid` representing a field of cherries where `grid[i][j]` represents the number of chocolates that you can collect from the `(i, j)` cell.

You have two robots that can collect chocolates for you:

- **Robot #1** is located at the **top-left corner** `(0, 0)`, and
- **Robot #2** is located at the **top-right corner** `(0, cols - 1)`.

Return the maximum number of chocolates collection using both robots by following the rules below:

- From a cell (i, j), robots can move to cell (i + 1, j - 1), (i + 1, j), or (i + 1, j + 1).
- When any robot passes through a cell, It picks up all chocolates, and the cell becomes an empty cell.
- When both robots stay in the same cell, only one takes the chocolates.
- Both robots cannot move outside of the grid at any moment.
- Both robots should reach the bottom row in grid.

Example:

Input:

r = 3, c = 4

grid = [[3,1,1],[2,5,1],[1,5,5],[2,1,1]]

Output:

24

Explanation:

Path of robot #1 and #2 are described in color green and blue respectively.

Cherries taken by Robot #1, (3 + 2 + 5 + 2) = 12. Cherries taken by Robot #2, (1 + 5 + 5 + 1) = 12. Total of cherries: 12 + 12 = 24.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **Solve()** which takes r rows, c column and a matrix grid and returns the maximum number of chocolates can be collect by two robots.

Expected Time Complexity: $O(r * c * c)$

Expected Space Complexity: $O(c * c * c)$

Constraint:

$2 \leq r \leq 70$

$0 \leq \text{grid}[i][j] \leq 100$

Count Lucky Permutations

Hard Accuracy: 0.0% Submissions: 1+ Points: 8

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given an array `arr[]` of integers having `N` elements and a non-weighted undirected graph having `N` nodes and `M` edges. The details of each edge in the graph is given to you in the form of list of list.

Your task is to find the number of lucky permutations of the given array.

An array permutation is said to be lucky if for every node V_i [$1 < i < N-1$] in the array there exists an edge between the nodes V_i and V_{i+1} in the given graph.

Example 1:

Input:

`N = 3, M = 2`

`arr = {1, 2, 3}`

`graph = {{3, 1}, {1, 2}}`

Output:

2

Explanation:

All possible permutations of the array are as follows-

`{1,2,3}`: There is an edge between 1 and 2 in the graph but not between 2 and 3.

`{2,1,3}`: There is an edge between (2,1) and (1,3) in the graph.

$\{3,1,2\}$: There is an edge between $(3,1)$ and $(1,2)$ in the graph.

Out of the 3 possible permutations, 2 are lucky. Therefore, answer is 2.

Example 2:

Input:

$n = 2, m = 1$

$arr = \{1, 1\}$

$graph = \{\{1, 2\}\}$

Output :

0

Explanation:

There is no lucky permutation in the given graph.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **luckyPermutations()** which takes the two integers **N** and **M**, an array **arr[]** and a list of lists named **graph** of size **M** as input parameters and returns the count of lucky permutations.

Expected Time Complexity: $O(N^2 \cdot 2^N)$

Expected Auxiliary Space: $O(N \cdot 2^N)$

Constraints:

$$2 \leq N \leq 15$$

$$1 \leq M \leq (N*(N-1))/2$$

$$1 \leq \text{arr}[i], \text{graph}[i][j] \leq N$$

There are no self loops and repeated edges in the graph.

Strongly connected component (Tarjans's Algo)

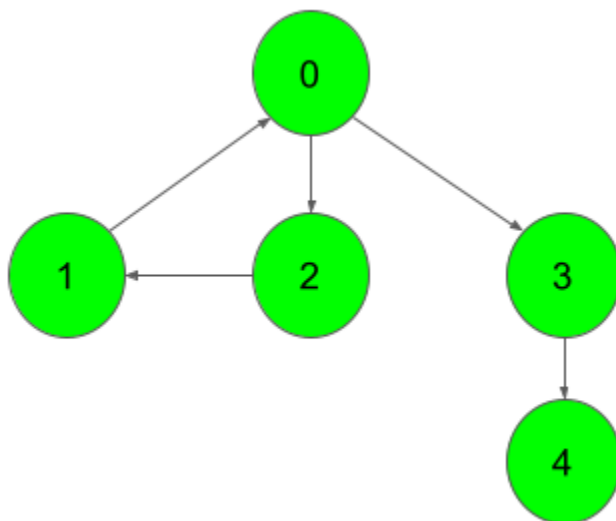
Hard Accuracy: **36.78%** Submissions: **16K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a Directed Graph with V vertices and E edges, Find the members of strongly connected components in the graph.

Example 1:

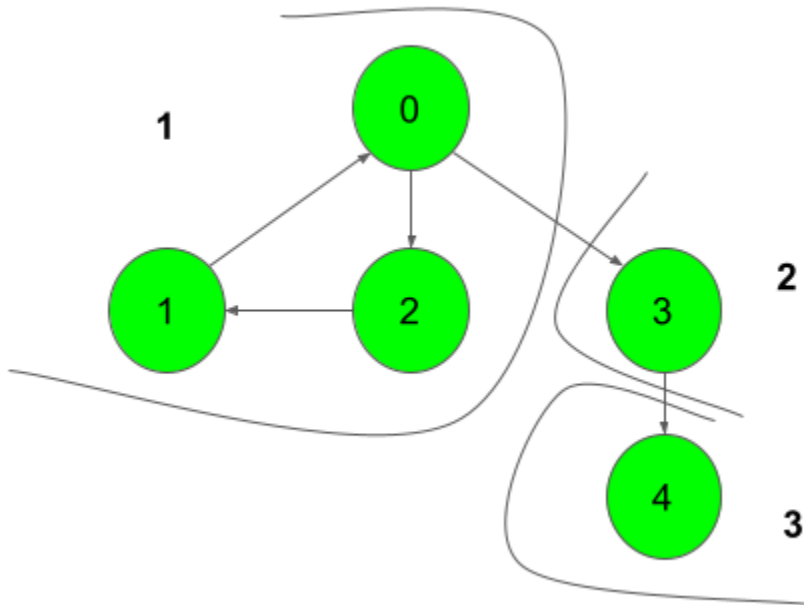
Input:



Output:

0 1 2,3,4,

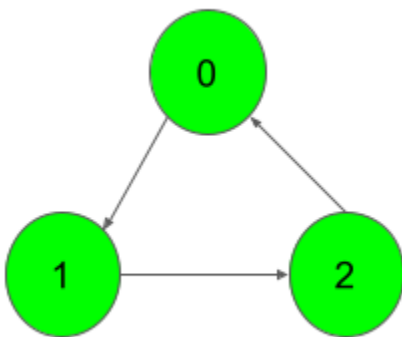
Explanation:



We can clearly see that there are 3 Strongly Connected Components in the Graph as mentioned in the Output.

Example 2:

Input:



Output:

0 1 2,

Explanation:

All of the nodes are connected to each other.

So, there's only one SCC as shown.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `tarjans()` which takes the number of vertices V and adjacency list of the graph as input parameters and returns a list of list of integers denoting the members of strongly connected components in the given graph.

Note: A single strongly connected component must be represented in the form of a list of integers sorted in the ascending order. The resulting list should consist of a list of all SCCs which must be sorted in a way such that a lexicographically smaller list of integers appears first.

Expected Time Complexity: $O(V + E)$.

Expected Auxiliary Space: $O(V)$.

Constraints:

$$1 \leq V \leq 10^5$$

$$1 \leq E \leq 10^5$$

$$0 \leq u, v \leq N-1$$

Word Ladder I

Hard Accuracy: **37.65%** Submissions: **16K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two distinct words `startWord` and `targetWord`, and a list denoting `wordList` of unique words of equal lengths. Find the length of the shortest transformation

sequence from startWord to targetWord.

Keep the following conditions in mind:

- A word can only consist of lowercase characters.
- Only one letter can be changed in each transformation.
- Each transformed word must exist in the wordList including the targetWord.
- startWord may or may not be part of the wordList

The second part of this problem can be found [here](#).

Note: If no possible way to transform sequence from startWord to targetWord **return 0**

Example 1:

Input:

```
wordList = {"des", "der", "dfr", "dgt", "dfs"}  
startWord = "der", targetWord= "dfs",
```

Output:

3

Explanation:

The length of the smallest transformation sequence from "der" to "dfs" is 3
i,e "der" -> "dfr" -> "dfs".

Example 2:

Input:

```
wordList = {"geek", "gefek"}  
startWord = "gedk", targetWord= "geek",
```

Output:

2

Explanation:

gedk -> geek

Example 3:

Input:

wordList = {"poon", "plee", "same", "poie", "plea", "plie", "poin"}

startWord = "toon", targetWord= "plea",

Output: 7

Explanation:

toon -> poon -> poin -> poie -> plie -> plee -> plea

Your Task:

You don't need to read or print anything, Your task is to complete the function **wordLadderLength()** which takes startWord, targetWord and wordList as input parameter and returns the length of the shortest transformation sequence from startWord to targetWord. If not possible return 0.

Expected Time Compelxity: $O(N^2 * M)$

Expected Auxiliary Space: $O(N * M)$ where N = length of wordList and $M = |wordList_i|$

Constraints:

$1 \leq N \leq 100$

$1 \leq M \leq 10$

Kill Captain America

Hard Accuracy: **48.45%** Submissions: **6K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Captain America is hiding from Thanos in a maze full of N rooms connected by M gates.

The maze is designed in such a way that each room leads to another room via gates. All connecting gates are unidirectional. Captain America is hiding only in those rooms which are accessible directly/indirectly through every other room in the maze.

Help Thanos find the number of rooms in which Captain America can hide.

Example 1:

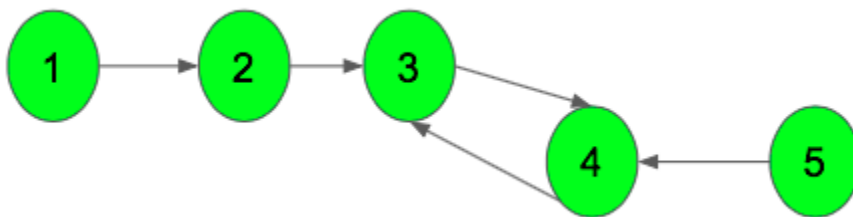
Input:

$N = 5$ and $M = 5$

$V = [[1, 2], [2, 3], [3, 4], [4, 3], [5, 4]]$

Output: 2

Explanation:



We can look closely after forming graph than captain america only can hide in a room 3 and 4 because they are the only room which have gates through them. So, answer is 2.

Example 2:**Input:**

$N = 2, M = 1$

$V = [[1, 2]]$

Output: 1

Your Task:

You don't need to read input or print anything. Your task is to complete the function **captainAmerica()** which takes the integer N , an integer M and 2-d array V as input parameters and returns the Total no of rooms.

Expected Time Complexity: $O(N+M)$

Expected Auxiliary Space: $O(N+M)$

Constraints:

$1 \leq n \leq 30000$

$1 \leq m \leq 200000$

$1 \leq p, q \leq n$

Articulation Point - II

Hard Accuracy: **62.66%** Submissions: **4K+** Points: **8**

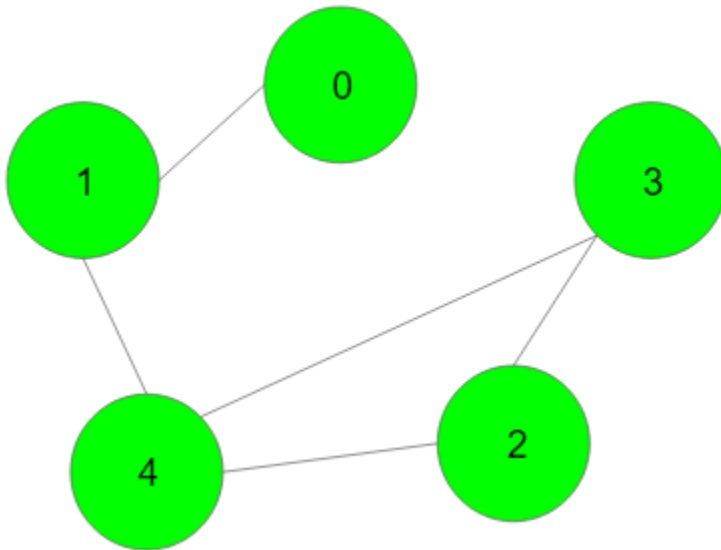
Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an undirected graph (not necessarily connected) with V vertices and adjacency list `adj`. You are required to find all the vertices removing which (and edges through it) disconnects the graph into 2 or more components, or in other words, removing which increases the number of connected components.

Note: Indexing is zero-based i.e nodes numbering from (0 to $V-1$). There might be loops present in the graph.

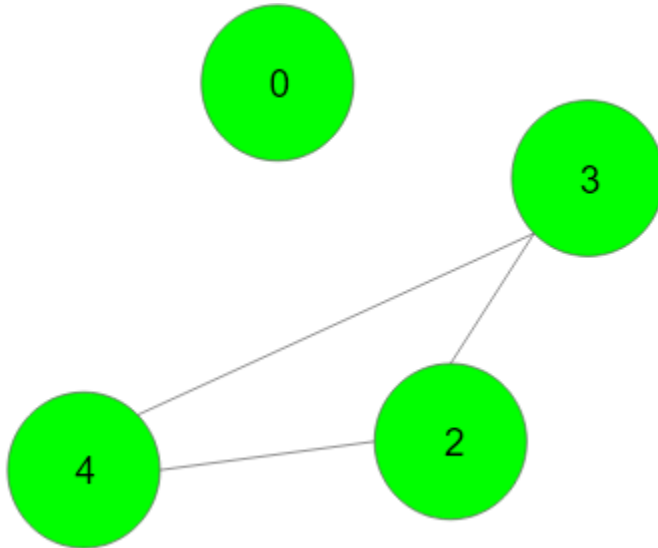
Example 1:

Input:

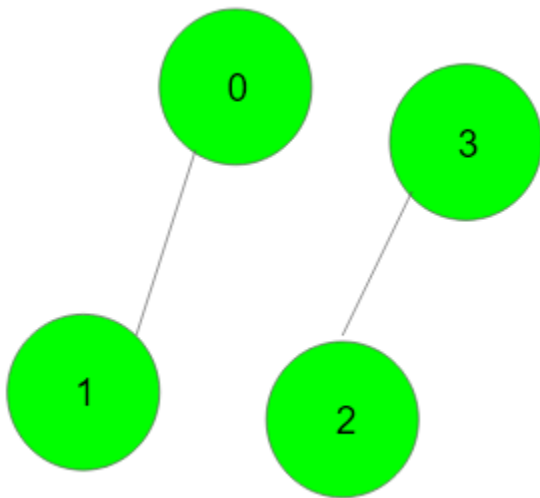


Output:{1,4}

Explanation: Removing the vertex 1 will disconnect the graph as-



Removing the vertex 4 will disconnect the graph as-



Your Task:

You don't need to read or print anything. Your task is to complete the function **articulationPoints()** which takes *V* and *adj* as input parameters and returns a list containing all the vertices removing which turn the graph into two or more disconnected components in sorted order. If there are no such vertices then returns a list containing -1.

Expected Time Complexity: $O(V + E)$

Expected Auxiliary Space: $O(V)$

Constraints:

$$1 \leq V \leq 10^4$$

Assignment Problem

Hard Accuracy: **30.31%** Submissions: **11K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are the head of a firm and you have to assign jobs to people. You have N persons working under you and you have N jobs that are to be done by these persons. Each person has to do exactly one job and each job has to be done by exactly one person. Each person has his own capability (in terms of time taken) to do any particular job. Your task is to assign the jobs to the persons in such a way that the total time taken is minimum. A job can be assigned to only one person and a person can do only one job.

Example 1:

Input:

$N = 2$

$Arr[] = \{3, 5, 10, 1\}$

Output:

4

Explanation:

The first person takes times 3 and 5 for jobs 1 and 2 respectively. The second

person takes times 10 and 1 for jobs 1 and 2 respectively. We can see that the optimal assignment will be to give job 1 to person 1 and job 2 to person 2 for a total for $3+1 = 4$.

Example 2:**Input:**

$N = 3$

$Arr[] = \{2, 1, 2, 9, 8, 1, 1, 1, 1\}$

Output:

3

Explanation:

The optimal arrangement would be to assign job 1 to person 3, job 2 to person 1 and job 3 to person 2.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `assignmentProblem()` which takes an Integer N and an array `Arr[]` of size N^2 as input and returns the time taken for the best possible assignment.

Expected Time Complexity: $O(N^2)$

Expected Auxiliary Space: $O(N^2)$

Constraints:

$1 \leq N \leq 30$

$1 \leq Arr[i] \leq 100$

Find the Maximum Flow

Hard Accuracy: **39.6%** Submissions: **7K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a graph which represents a flow network with N vertices numbered 1 to N and M edges. Find the maximum flow from vertex numbered 1 to vertex numbered N.

In a flow network, every edge has a flow capacity and the maximum flow of a path can't exceed the flow-capacity of an edge in the path.

Example 1:

Input:

N = 5, M = 4

Edges[] = {{1,2,1},{3,2,2},{4,2,3},{2,5,5}}

Output: 1

Explanation:

1 - 2 - 3

/ \

4 5

1 unit can flow from 1 -> 2 -> 5

Example 2:

Input:

N = 4, M = 4

Edges[] = {{1,2,8},{1,3,10},{4,2,2},{3,4,3}}

Output: 5

Explanation:

1 - 2

| |

3 - 4

3 unit can flow from 1 -> 3 -> 4

2 unit can flow from 1 -> 2 -> 4

Total max flow from 1 to N = 3+2=5

Your Task:

You don't need to read input or print anything. Your task is to complete the function `solve()` which takes the `N` (the number of vertices), `M` (the number of Edges) and the array `Edges[]` (Where `Edges[i]` denoting an undirected edge between `Edges[i][0]` and `Edges[i][1]` with a flow capacity of `Edges[i][2]`), and returns the integer denoting the maximum flow from 1 to N.

Expected Time Complexity: $O(\text{max_flow} * M)$

Expected Auxiliary Space: $O(N + M)$

Where `max_flow` is the maximum flow from 1 to N

Constraints:

$1 \leq N, M, \text{Edges}[i][2] \leq 1000$

$1 \leq \text{Edges}[i][0], \text{Edges}[i][1] \leq N$

Find minimum s-t cut in a flow network

Hard Accuracy: **21.0%** Submissions: **1K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a weighted graph of `N` vertices numbered from `0` to `N-1` in the form of adjacency matrix `A[][]` and two integers `S` denoting the number of source vertex and `T` denoting the number of sink vertex. The task is to find minimum capacity s-t cut of the given network. An s-t cut is a cut that requires the source node 'S' and the sink node 'T' to be in different subsets, and it consists of edges

going from the source's side to the sink's side. The capacity of an s-t cut is defined by the sum of the capacity of each edge in the cut-set. In other words, you have to find out all the edges which has to be removed to make it impossible to reach the sink node from source node, and the edges you select should have a minimum sum of weights. You have to return all the edges included in the minimum capacity s-t cut and if there are no edges in minimum capacity s-t cut, return "-1".

Example 1:**Input:**

N = 2

 $A[][] = [[0, 3],$
 $[0, 0]]$

S = 0

T = 1

Output:

0 1

Explanation: We have to remove the edge going from 0th vertex to 1st vertex.**Example 2:****Input:**

N = 5

 $A[][] = [[0, 0, 0, 0, 0],$
 $[0, 0, 2, 3, 0],$
 $[0, 0, 0, 0, 0],$
 $[0, 0, 0, 0, 0],$
 $[0, 0, 0, 0, 0]]$

S = 0

$T = 4$

Output:

-1

Explanation: There are no edges in minimum capacity s-t cut.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `minimumCut()` which takes the adjacency matrix `A[][]`, source node number `S`, sink node number `T` and number of vertices `N` and returns a list of integers `res[]` where `res[2*i-1]` and `res[2*i]` denotes an edge in minimum capacity s-t cut where $1 \leq i \leq \text{length}(\text{res})/2$, if there are no edges in minimum capacity s-t cut, return only one integer "-1" in `res[]`.

Expected Time Complexity: $O(\text{max_flow} * N^2)$

Expected Auxiliary Space: $O(N^2)$

Constraints:

$1 \leq N \leq 50$

$0 \leq S, T < N$

Critical Connections

Hard Accuracy: **18.89%** Submissions: **3K+** Points: **8**

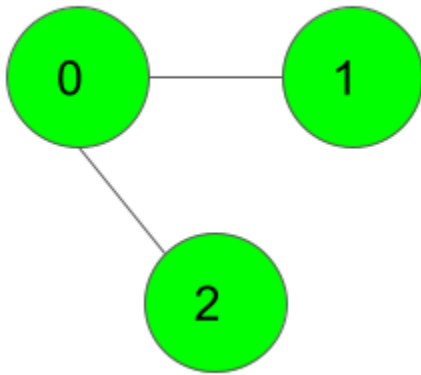
Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an undirected Connected graph of V vertices and E edges.

A critical connection is an edge that, if removed, will make some nodes unable to reach some other nodes. Find all critical connections in the graph.

Note: There are many possible orders for the answer. You are supposed to print

the edges in sorted order, and also an edge should be in sorted order too. So if there's an edge between node 1 and 2, you should print it like (1,2) and not (2,1).

Example 1:**Input:****Output:**

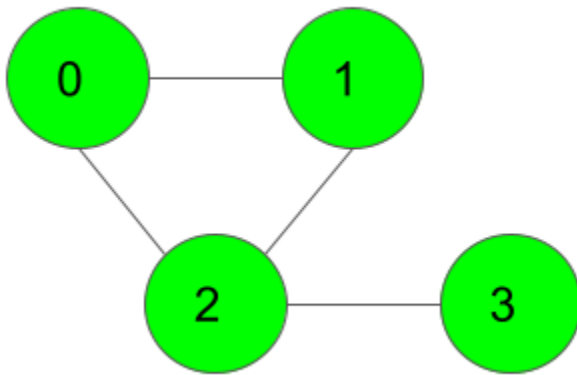
0 1

0 2

Explanation:

Both the edges in the graph are Critical connections.

Example 2:**Input:**



Output:

2 3

Explanation:

The edge between nodes 2 and 3 is the only Critical connection in the given graph.

Your task:

You don't need to read input or print anything. Your task is to complete the function **criticalConnections()** which takes the integer V denoting the number of vertices and an adjacency list **adj** as input parameters and returns a list of lists containing the Critical connections in the sorted order.

Expected Time Complexity: $O(V + E)$

Expected Auxiliary Space: $O(V)$

Constraints:

$1 \leq V, E \leq 10^5$

Word Ladder II

Hard Accuracy: **40.62%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two distinct words **startWord** and **targetWord**, and a list denoting **wordList** of unique words of equal lengths. Find all shortest transformation sequence(s) from **startWord** to **targetWord**. You can return them in any order possible.

Keep the following conditions in mind:

- A word can only consist of lowercase characters.
- Only one letter can be changed in each transformation.
- Each transformed word must exist in the wordList including the **targetWord**.
- **startWord** may or may not be part of the wordList.
- Return an empty list if there is no such transformation sequence.

The first part of this problem can be found [here](#).

Example 1:

Input:

startWord = "der", **targetWord** = "dfs",
wordList = {"des", "der", "dfr", "dgt", "dfs"}

Output:

der dfr dfs
der des dfs

Explanation:

The length of the smallest transformation is 3.
And the following are the only two ways to get to **targetWord**:-

```
"der" -> "des" -> "dfs".  
"der" -> "dfr" -> "dfs".
```

Example 2:

Input:

```
startWord = "gedk", targetWord = "geek",  
wordList = {"geek", "gefk"}
```

Output:

```
"gedk" -> "geek"
```

Your Task:

You don't need to read or print anything, Your task is to complete the function `findSequences()` which takes `startWord`, `targetWord` and `wordList` as input parameter and returns a list of list of strings of the shortest transformation sequence from `startWord` to `targetWord`.

Note: You don't have to return -1 in case of no possible sequence. Just return the Empty List.

Expected Time Compelxity: $O(N * (\log N * M * 26))$

Expected Auxiliary Space: $O(N * M)$ where N = length of `wordList` and M = $|\text{wordList}_i|$

Constraints:

$1 \leq N \leq 100$

$1 \leq M \leq 10$

Articulation Point - I

Hard Accuracy: **39.26%** Submissions: **22K+** Points: **8**

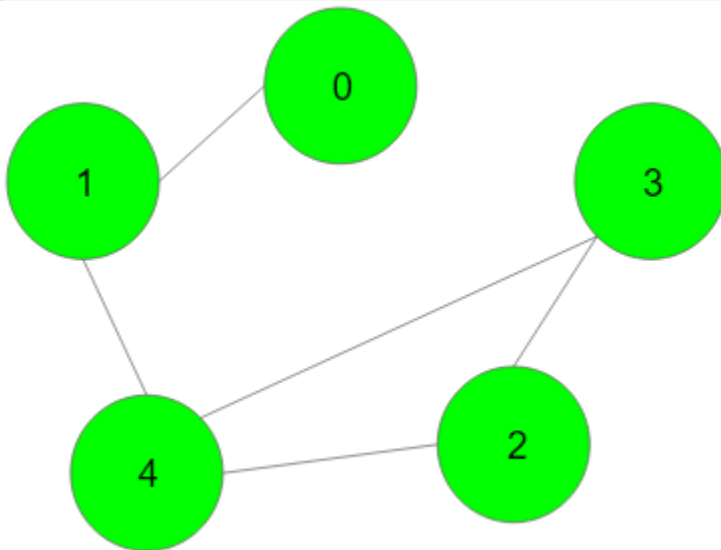
Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an undirected connected graph with V vertices and adjacency list **adj**. You are required to find all the vertices removing which (and edges through it) disconnects the graph into 2 or more components.

Note: Indexing is zero-based i.e nodes numbering from (0 to $V-1$). There might be loops present in the graph.

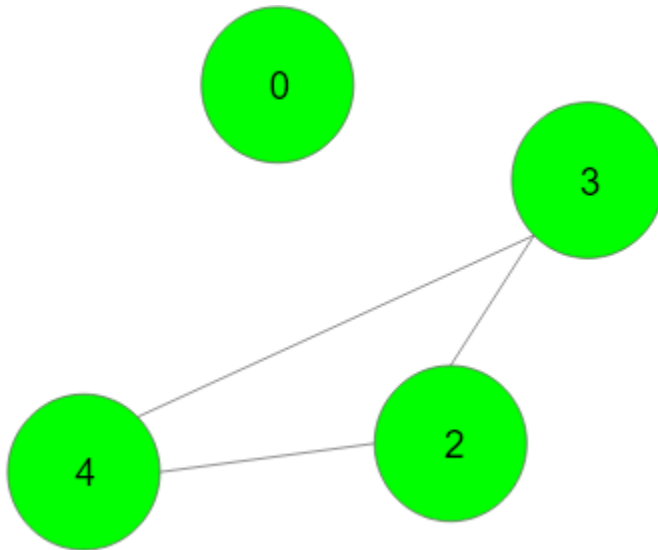
Example 1:

Input:

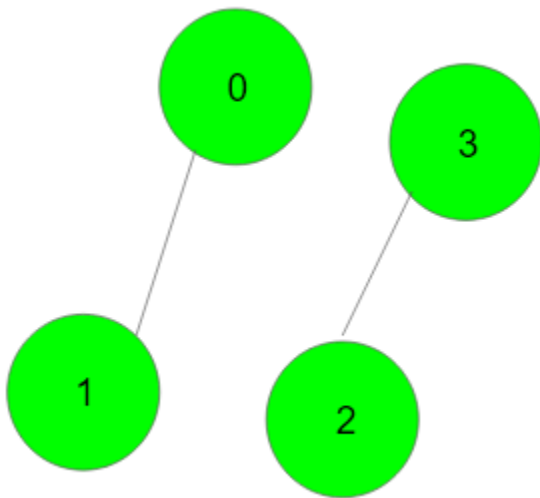


Output:{1,4}

Explanation: Removing the vertex 1 will disconnect the graph as-



Removing the vertex 4 will disconnect the graph as-



Your Task:

You don't need to read or print anything. Your task is to complete the function **articulationPoints()** which takes **V** and **adj** as input parameters and returns a list containing all the vertices removing which turn the graph into two or more disconnected components in sorted order. If there are no such vertices then returns a list containing -1.

Expected Time Complexity: $O(V + E)$

Expected Auxiliary Space: $O(V)$

Constraints:

$1 \leq V \leq 10^5$

Geek in a Maze

Hard Accuracy: **45.9%** Submissions: **6K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Geek is in a maze of size $N * M$. Each cell in the maze is made of either '.' or '#'. An empty cell is represented by '.' and an obstacle is represented by '#'. If Geek starts at cell (R, C) , find how many different empty cells he can pass through while avoiding the obstacles. He can move in any of the four directions but he can move up at most U times and he can move down at most D times.

Example 1:

Input:

$N = 3, M = 3$

$R = 1, C = 0$

$U = 1, D = 1$

$\text{mat} = \{\{'.', ' ', ' '\},$
 $\{'.', '#', ' '\},$
 $\{'#', ' ', ' '\}\}$

Output: 5

Explanation: Geek can reach
(1, 0), (0, 0), (0, 1), (0, 2), (1, 2)

Example 2:

Input:

N = 3, M = 4

R = 1, C = 0

U = 1, D = 2

```
mat = {{'.', '.', '.'},
        {'.', '#', '.'},
        {'.', '.', '.'},
        {'#', '.', '.'}}
```

Output: 10

Explanation: Geek can reach all the
cells except for the obstacles.

Your Task:

You don't need to read input or print anything. Complete the function **numberOfCells()** which takes **N, M, R, C, U, D** and the two dimensional character array **mat[][]** as input parameters and returns the number of cells geek can visit(If he is standing on obstacle he can not move).

Constraints:

$1 \leq N * M \leq 10^6$

$mat[i][j] = \text{'#'} \text{ or } \text{'.'}$

$0 \leq R \leq N-1$

$0 \leq C \leq M-1$

Phone directory

Hard Accuracy: **25.68%** Submissions: **34K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a list of contacts **contact[]** of length **n** where each contact is a string which exist in a phone directory and a query string **s**. The task is to implement a search query for the phone directory. Run a search query for each prefix **p** of the query string **s** (*i.e.* from index 1 to **|s|**) that prints all the distinct contacts which have the same prefix as **p** in **lexicographical increasing order**. Please refer the explanation part for better understanding.

Note: If there is no match between query and contacts, print "0".

Example 1:

Input:

n = 3

contact[] = {"geeikistest", "geeksforgeeks",
"geeksfortest"}

s = "geeips"

Output:

geeikistest geeksforgeeks geeksfortest

geeikistest geeksforgeeks geeksfortest

geeikistest geeksforgeeks geeksfortest

geeikistest

0

0

Explanation: By running the search query on contact list for "g" we get: "geeikistest",

"geeksforgeeks" and "geeksfortest".

By running the search query on contact list for "ge" we get: "geeikistest" "geeksforgeeks" and "geeksfortest".

By running the search query on contact list for "gee" we get: "geeikistest" "geeksforgeeks" and "geeksfortest".

By running the search query on contact list for "geei" we get: "geeikistest".

No results found for "geeip", so print "0".

No results found for "geeips", so print "0".

Your Task:

You do not need to read input or print anything. Your task is to complete the function **displayContacts()** which takes **n**, **contact[]** and **s** as input parameters and returns a list of list of strings for required prefixes. If some prefix has no matching contact return "0" on that list.

Expected Time Complexity: $O(|s| * n * \max|\text{contact}[i]|)$

Expected Auxiliary Space: $O(n * \max|\text{contact}[i]|)$

Constraints:

$1 \leq T \leq 100$, T = number of test cases

$1 \leq n \leq 50$

$1 \leq |\text{contact}[i]| \leq 50$

$1 \leq |s| \leq 6$

Word Break (Trie)

Hard Accuracy: **32.32%** Submissions: **7K+** Points: **8**

Given a string **A** and a dictionary of **n** words **B**, find out if **A** can be segmented into a space-separated sequence of dictionary words.

Example 1:

Input:

`n = 12`

`B = { "i", "like", "sam", "sung", "samsung",
"mobile", "ice", "cream", "icecream", "man",
"go", "mango" }, A = "ilike"`

Output: 1

Explanation: The string can be segmented as "i like".

Example 2:

Input:

`n = 12`

`B = { "i", "like", "sam", "sung", "samsung",
"mobile", "ice", "cream", "icecream", "man",
"go", "mango" }, A = "ilikesamsung"`

Output: 1

Explanation: The string can be segmented as "i like samsung" or "i like sam sung".

Your Task:

Complete **wordBreak()** function which takes a string and list of strings as a parameter and returns 1 if it is possible to break words, else return 0. You don't need to read any input or print any output, it is done by driver code.

Expected time complexity: $O(n \cdot l + |A|^2)$ where l is the length of longest string present in the dictionary and $|A|$ is the length of string A

Expected auxiliary space: $O(|A| + k)$, where k = sum of length of all strings present in B

Constraints:

$1 \leq N \leq 12$

$1 \leq s \leq 1000$, where s = length of string A

The length of each word is less than 15.

Palindrome Pairs

Hard Accuracy: **27.71%** Submissions: **12K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array of strings `arr[]` of size N , find if there exists 2 strings `arr[i]` and `arr[j]` such that `arr[i]+arr[j]` is a palindrome i.e the concatenation of string `arr[i]` and `arr[j]` results into a palindrome.

Example 1:

Input:

$N = 6$

`arr[] = {"geekf", "geeks", "or", "keeg", "abc", "bc"}`

Output: 1

Explanation: There is a pair "geekf" and "keeg".

Example 2:

Input:

N = 5

arr[] = {"abc", "xyxcba", "geekst", "or", "bc"}

Output: 1

Explanation: There is a pair "abc" and "xyxcba".

Your Task:

You don't need to read input or print anything. Your task is to complete the function **palindromePair()** which takes the array arr[], its size N and returns true if palindrome pair exists and returns false otherwise.

The driver code itself prints 1 if returned value is true and prints 0 if returned value is false.

Expected Time Complexity: $O(N \cdot l^2)$ where l = length of longest string in arr[]

Expected Auxiliary Space: $O(N \cdot l^2)$ where l = length of longest string in arr[]

Constraints:

$1 \leq N \leq 2 \cdot 10^4$

$1 \leq |arr[i]| \leq 10$

Shortest Unique prefix for every word

Hard Accuracy: **52.55%** Submissions: **9K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array of words, find all shortest unique prefixes to represent each word in the given array. Assume that no word is prefix of another.

Example 1:

Input:

N = 4

arr[] = {"zebra", "dog", "duck", "dove"}

Output: z dog du dov

Explanation:

z => zebra

dog => dog

duck => du

dove => dov

Example 2:

Input:

N = 3

arr[] = {"geeksgeeks", "geeksquiz",
 "geeksforgeeks"};

Output: geeksg geeksq geeksf

Explanation:

geeksgeeks => geeksg

geeksquiz => geeksq

geeksforgeeks => geeksf

Your task:

You don't have to read input or print anything. Your task is to complete the function **findPrefixes()** which takes the array of strings and it's size N as input and returns a list of shortest unique prefix for each word

Expected Time Complexity: $O(N \times \text{length of each word})$

Expected Auxiliary Space: $O(N \times \text{length of each word})$

Constraints:

$1 \leq N$, Length of each word ≤ 1000

Points in Straight Line

Hard Accuracy: **18.37%** Submissions: **7K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given two arrays **X[]** and **Y[]** of points where (X_i, Y_i) represents a point on the X-Y plane. Your task is to complete the function **maxPoints** which returns an integer denoting the maximum number of points that lie on the same straight line.

Example 1:

Input:

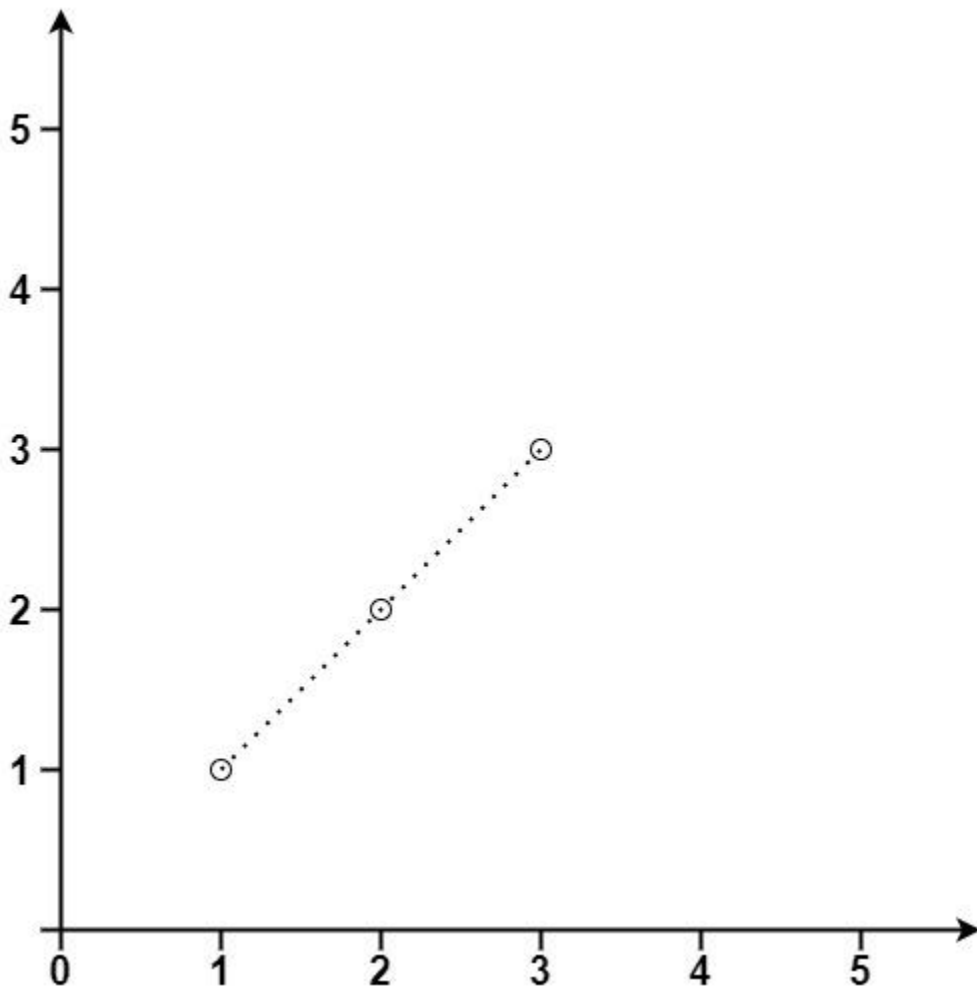
X[] = {1, 2, 3}

Y[] = {1, 2, 3}

Output: 3

Explanation:

The points are (1,1), (2,2) and (3,3).



Example 2:

Input:

$X[] = \{1, 3, 5, 4, 2, 1\}$

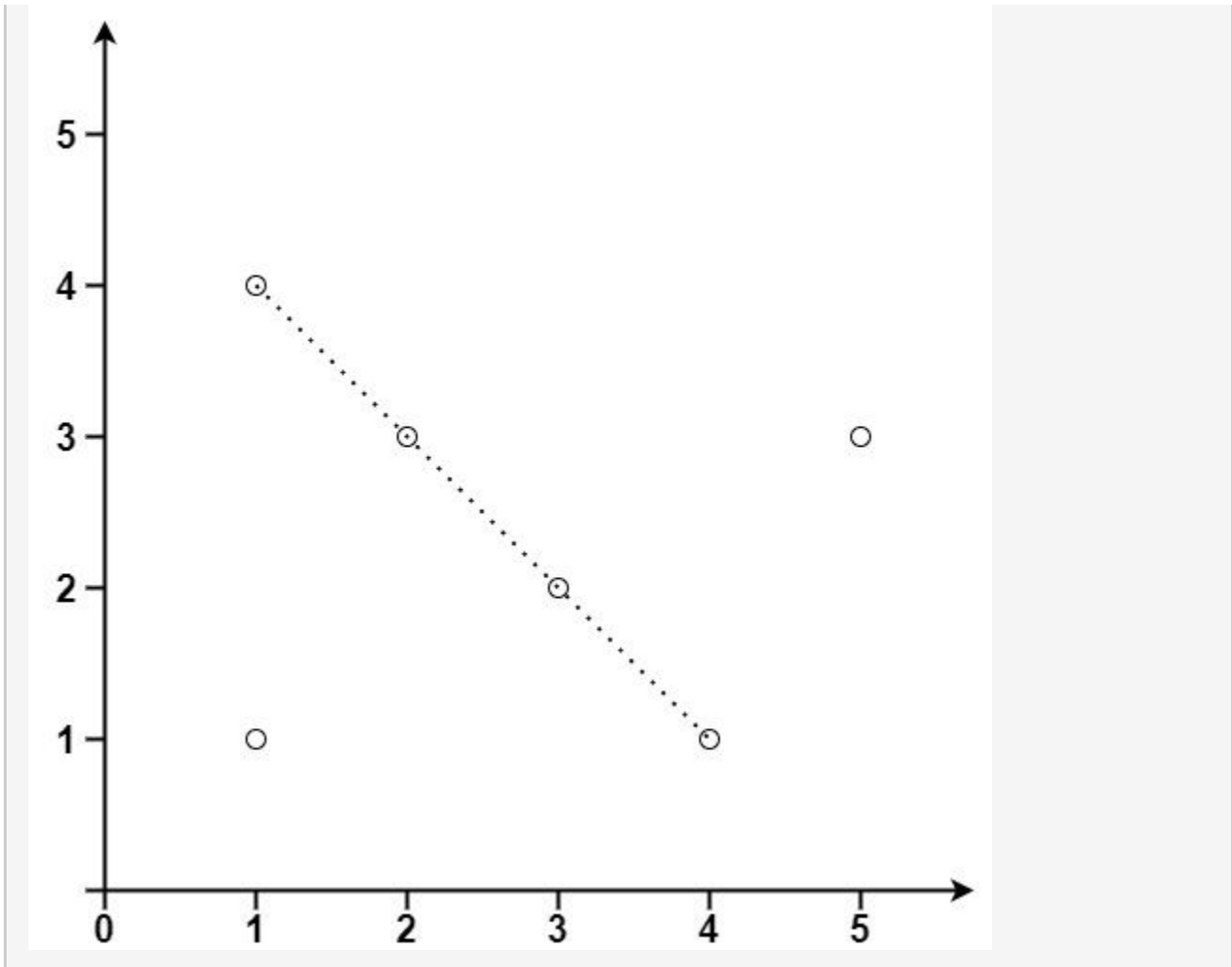
$Y[] = \{1, 2, 3, 1, 3, 4\}$

Output: 4

Explanation:

The points are-

(1,1),(3,2),(5,3),(4,1),(2,3),(1,4)

**Your Task:**

You don't need to read input or print anything. Your task is to complete the function **maxPoints()** which takes two lists of coordinates as input and returns the maximum number of points that lies on the same line.

Expected Time Complexity: $O(N^2)$

Expected Auxiliary Space: $O(N)$

Constraints:

$$1 \leq N \leq 300$$

$$-10^4 \leq x_i, y_i \leq 10^4$$

Very Tough Problem

Hard Accuracy: **45.32%** Submissions: **1K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given three numbers N , S and X . You have to tell whether it is possible to construct such sequence A of length N , where each $A_i \geq 0$ for $1 \leq i \leq N$ and the sum of all numbers in a sequence is equal to S , and the **XOR** of sequence equals to X .

Example 1:

Input:

$$N = 2$$

$$S = 10$$

$$X = 0$$

Output: Yes

Explanation:

We can use the sequence "5 5", because the XOR of all elements will be 0, and sum 10

Example 2:

Input:

$$N = 1$$

$$S = 3$$

$X = 2$

Output: No

Your Task:

You don't need to read input or print anything. Your task is to complete the function **toughProblem()** which takes the integer N, integer S, and integer X as input parameters and returns the “Yes” (without quotes) if it is possible to have such a sequence and “No” if it is not possible to have such a sequence.

Expected Time Complexity: $O(1)$

Expected Auxiliary Space: $O(1)$

CONSTRAINTS

$1 \leq N \leq 10^5$

$0 \leq S, X \leq 10^9$

Nth Natural Number

Hard Accuracy: **29.99%** Submissions: **43K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a positive integer N. You have to find N^{th} natural number after removing all the numbers containing digit 9.

Example 1:

Input:

N = 8

Output:

8

Explanation:

After removing natural numbers which contains digit 9, first 8 numbers are 1,2,3,4,5,6,7,8 and 8th number is 8.

Example 2:**Input:**

N = 9

Output:

10

Explanation:

After removing natural numbers which contains digit 9, first 9 numbers are 1,2,3,4,5,6,7,8,10 and 9th number is 10.

Your Task:

You don't need to read input or print anything. Complete the function **findNth()** which accepts an integer N as input parameter and return the Nth number after removing all the numbers containing digit 9.

Expected Time Complexity: $O(\log N)$

Expected Auxiliary Space: $O(1)$

Constraints:

$$1 \leq N \leq 10^{12}$$

Return two prime numbers**Hard** Accuracy: **58.64%** Submissions: **29K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an even number N (greater than 2), return two prime numbers whose sum will be equal to given number. There are several combinations possible. Print only the pair whose minimum value is the smallest among all the minimum values of pairs and print the minimum element first.

NOTE: A solution will always exist, read [Goldbachs conjecture](#).

Example 1:**Input:** $N = 74$ **Output:** 3 71

Explanation: There are several possibilities like 37 37. But the minimum value of this pair is 3 which is smallest among all possible minimum values of all the pairs.

Example 2:**Input:** 4**Output:** 2 2

Explanation: This is the only possible

partitioning of 4.

Your Task:

You do not need to read input or print anything. Your task is to complete the function `primeDivision()` which takes N as input parameter and returns the partition satisfying the condition.

Expected Time Complexity: $O(N \cdot \log(\log N))$

Expected Auxiliary Space: $O(N)$

Constraints:

$$4 \leq N \leq 10^4$$

Bike Racing

Hard Accuracy: **49.72%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Geek is organising a bike race with N bikers. The initial speed of the i th biker is denoted by H_i Km/hr and the acceleration of i th biker as A_i Km/Hr². A biker whose speed is 'L' or more, is considered be a fast biker. The total speed on the track for every hour is calculated by adding the speed of each fast biker in that hour. When the total speed on the track is 'M' kilometers per hour or more, the safety alarm turns on.

Find the minimum number of hours after which the safety alarm will start.

Example 1:**Input:**

$N = 3, M = 400, L = 120$

$H = \{20, 50, 20\}$

$A = \{20, 70, 90\}$

Output: 3

Explanation:

Speeds of all the Bikers at ith hour

Biker1= [20 40 60 80 100]

Biker2= [50 120 190 260 330]

Biker3= [20 110 200 290 380]

Initial Speed on track = 0

because none of the biker's speed is fast enough.

Speed on track after 1st Hour= 120

Speed on track after 2nd Hour= 190+200=390

Speed on track after 3rd Hour= 260+290=550

Alarm will start at 3rd Hour.

Example 2:

Input:

$N = 2, M = 60, L = 120$

$H = \{50, 30\}$

$A = \{20, 40\}$

Output: 3

Explanation:

Speeds of all the Bikers at ith hour

Biker1= [50 70 90 110 130]

Biker2= [30 70 110 150 190]

Initial Speed on track = 0 because none of the biker's speed is fast enough.

Speed on track at 1st Hour= 0

Speed on track at 2nd Hour= 0
Speed on track at 3rd Hour= 150
Alarm will buzz at 3rd Hour.

Your Task:

You do not need to read input or print anything. Your task is to complete the function **buzzTime()** which takes N, M, L and array H and array A as input parameters and returns the time when alarm buzzes.

Expected Time Complexity: $O(N \cdot \log(\max(L, M)))$

Expected Auxiliary Space: $O(1)$

Constraints:

$$1 \leq N \leq 10^5$$

$$1 \leq L, M \leq 10^{10}$$

$$1 \leq H_i, A_i \leq 10^9$$

$nCr \bmod M$ | Part 1

Hard Accuracy: **23.67%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given 2 integers n and r. You task is to calculate $nCr \% 1000003$.

Example 1:

Input: n = 5, r = 2

Output: 10

Explanation: ${}^5C_2 = 5! / (2! * 3!) = 10$

Example 2:

Input: n = 3, r = 2

Output: 3

Explanation: ${}^3C_2 = 3! / (2! * 1!) = 3$

Your Task:

You don't need to read or print anything. Your task is to complete the function **nCr()** which takes n and r as input parameter and returns nCr modulo 1000003.

Expected Time Complexity: $O(m * \log_m n)$ where $m = 1000003$

Expected Space Complexity: $O(m)$

Constraints:

$1 \leq n \leq r \leq 10^{16}$

Convex Hull

Hard Accuracy: **9.41%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Convex Hull of a set of points, in 2D plane, is a convex polygon with minimum area such that each point lies either on the boundary of polygon or inside it. Now

given a set of points the task is to find the convex hull of points.

Example 1:

Input: points_list = {{1,2},{3,1},{5,6}}

Output: {{1,2},{3,1},{5,6}}

Example 2:

Input : points_list = {{5,1},{4,4},{1,2}}

Output: {{1,2},{4,4},{5,1}}

Your Task:

You don't need to read or print anything. Your task is to complete the function **FindConvexHull()** which takes points_list as input parameter and returns Convex Hull of given points in a list. If not possible returns a list containing -1.

Expected Time Complexity: $O(n \log(n))$

Expected Space Complexity: $O(n)$ where n = total no. of points

Constraints:

$1 \leq n \leq 10^4$

$-10^5 \leq x, y \leq 10^5$

Next Optimal Move in Tic Tac Toe

Hard Accuracy: **22.96%** Submissions: **784+** Points: **8**

You are given a middle game situation of the game [Tic Tac Toe](#). It is given that it is player "X's" turn and you need to give to most optimal position for the turn. The situation is given as a 3 x 3 character matrix **board**. '_' refers to the place is empty. 'o' refers that player O marked it in his turn at some time and 'x' refers that player X marked it in his turn at some time. It is player X's turn. Tell him the most optimal solution.(Assume player O played first).

Example 1:

Input: board = {{o, _ ,_},
 { _ , _ ,_},
 { _ , _ ,_}}

Output: 1 1

Explanation: Placing a 'x' in the (1, 1) that is the center of the board is the most optimal approach for x.

Your Task:

You do not need to read input or print anything. Your task is to complete the function **findBestMove()** which takes board as input parameter and returns the best optimal move in a list where the first one is the row index and the second one is the column index.

Expected Time Complexity: $O(2^9)$

Expected Auxiliary Space: $O(1)$

Constraints:

board[i][j] = 'o' / 'x' / '_'

Closest Palindrome

Hard Accuracy: **9.67%** Submissions: **40K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a number num, our task is to find the closest Palindrome number whose absolute difference with given number is minimum. If 2 Palindrome numbers have same absolute difference from the given number, then find the smaller one.

Example 1:

Input: num = 9

Output: 9

Explanation: 9 itself is a palindrome number.

Example 2:

Input: num = 489

Output: 484

Explanation: closest palindrome numbers from 489 are 484 and 494. Absolute difference between 489 and 494 is equal to the absolute difference between 484 and 489 but 484 is smaller than 494.

Your Task:

You don't need to read or print anything. Your task is to complete the function **closestPalindrome()** which takes num as input parameter and returns the closest palindrome.

Expected Time Complexity: $O(\log_{10} \text{num})$

Expected Space Complexity: $O(1)$

Constraints:

$1 \leq \text{num} \leq 10^{14}$

Find the pattern

Hard Accuracy: **15.63%** Submissions: **837+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a sequence as follow:

1,1,7,19,73....

Find the N^{th} term of the given sequence.

Example 1:

Input: $N = 1$

Output: 1

Example 2:

Input: $N = 4$

Output: 19

Your Task:

Your task is to complete the function **NthTerm()** which takes N as input paramater

and returns N^{th} term of the given sequence modulo 10^9+7 .

Expected Time Complexity: $O(\log(N))$

Expected Space Complexity: $O(K)$ where K is constant.

Constraints:

$1 \leq N \leq 10^{10}$

Akku and Arrays

Hard Accuracy: **25.57%** Submissions: **887+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Akku have solved many problems, she is genius. One day her friend gave her an Array of size n and asked her to perform some queries of following type:

Each query consists of three integers

1 A B : Update the Array at index A by value B

2 A B : if the subarray from index A to B (both inclusive) is

1. Both increasing(Non-decreasing) and decreasing(Non-increasing) print -1
2. Only increasing(Non-decreasing) print 0
3. Only decreasing(Non-increasing) print 1
4. Neither increasing nor decreasing print -1

Akku needs your help, can you help her.

Example 1:

Input: nums = {1,5,7,4,3,5,9},

Queries = {{2,1,3},{1,7,4},{2,6,7}}

Output: {0,1}

Explanation: For the 1st query given :

A = 1, B = 3. From 1 to 3(1,5,7) elements are in increasing order. So answer is 0.

For the 2nd query we have to update the 7th element of the array by 4. So new updated array will be {1,5,7,4,3,5,4}

For the 3rd query A = 6, B = 7. From 6 to 7 (5, 4) elements are in descending order. So answer is 1.

Your Task:

You don't need to read or print anything. Your task is to complete the function **solveQueries()** which takes nums and Queries as input parameter and returns a list containing the answer for the 2nd type of query.

Expected Time Complexity: $O(n \cdot \log(n))$

Expected Space Complexity: $O(n)$

Constraints:

$1 \leq n \leq 10^4$

$1 \leq \text{nums}[i] \leq 10^4$

$1 \leq \text{No. of queries} \leq 10^4$

Interesting Queries

Hard Accuracy: **11.83%** Submissions: **5K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `nums` of `n` elements and `q` queries . Each query consists of two integers `l` and `r` . Your task is to find the number of elements of `nums[]` in range `[l,r]` which occur at least `k` times.

Example 1:

Input: `nums = {1,1,2,1,3}`, `Queries = {{1,5}, {2,4}}`, `k = 1`

Output: `{3,2}`

Explanation: For the 1st query, from `l=1` to `r=5` 1, 2 and 3 have the frequency at least 1.

For the second query, from `l=2` to `r=4`, 1 and 2 have the frequency at least 1.

Example 1:

Input: `nums = {1,2,3,1}`, `Queries = {{1,4}, {2,4},{4,4}}`, `k = 2`

Output: `{1,0,0}`

Explanation: For the 1st query, from `l=1` to `r=4` 1 has the frequency at least 2.

For the second query, from `l=2` to `r=4`, no number has the frequency at least 2.

For the third query, from `l=4` to `r=4`, no number has

the frequency atleast 2.

Your Task:

Your task is to complete the function **solveQueries()** which takes **nums**, **Queries** and **k** as input parameter and returns a list containing the answer for each query.

Expected Time Complexity: $O(n \cdot \sqrt{n} \cdot \log(n))$

Expected Space Complexity: $O(n)$

Constraints:

$1 \leq n$, no of Queries, $k \leq 10^4$

$1 \leq \text{nums}[i] \leq 10^3$

$1 \leq \text{Queries}[i][0] \leq \text{Queries}[i][1] \leq n$

Hexadecimal to decimal counter

Hard Accuracy: **14.63%** Submissions: **704+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given a number **str**(in string format) in hexadecimal. A new number can be made from the number **str** by selecting any subsequence of it (in HexaDecimal) and rearranging it.

You have to tell the number of distinct numbers that can be made from number **n**.

Example 1:

Input: str = "1F"

Output: 4

Explanation: For 1F possible combination are
1, F, 1F, F1.

Example 2:

Input: str = "1FF"

Output: 8

Explanation: For 1FF possible combinations are
1, F, 1F, F1, FF, 1FF, F1F, FF1.

Your Task:

You don't need to read or print anything. Your task is to complete the function **countOfDistinctNo()** which takes str in string format as input and returns the total possible combination modulo 10^9+7 .

Expected Time Complexity: $O(L*L)$ where L is the length of the string str.

Expected Space Complexity: $O(L)$

Constraints:

$1 \leq n \leq 2^{8000}$ where n = str in decimal.

Hexadecimal to decimal counter

Hard Accuracy: **14.63%** Submissions: **704+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given a number str(in string format) in hexadecimal. A new number can be made from the number str by selecting any subsequence of it (in HexaDecimal) and rearranging it.

You have tell the number of distinct numbers that can be made from number n.

Example 1:

Input: str = "1F"

Output: 4

Explanation: For 1F possible combination are
1, F, 1F, F1.

Example 2:

Input: str = "1FF"

Output: 8

Explanation: For 1FF possible combinations are
1, F, 1F, F1, FF, 1FF, F1F, FF1.

Your Task:

You don't need to read or print anything. Your task is to complete the function **countOfDistinctNo()** which takes str in string format as input and returns the total possible combination modulo 10^9+7 .

Expected Time Complexity: $O(L*L)$ where L is the length of the string str.

Expected Space Complexity: $O(L)$

Constraints:

$1 \leq n \leq 2^{8000}$ where n = str in decimal.

Create your own Calculator

Hard Accuracy: **38.3%** Submissions: **1K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given the co-efficients of X,Y and Z in a system of simultaneous equations. Find the values of X,Y and Z.

$$a_1X + b_1Y + c_1Z = d_1$$

$$a_2X + b_2Y + c_2Z = d_2$$

$$a_3X + b_3Y + c_3Z = d_3$$

Example 1:

Input:

Arr = {{5, 4, 2, 0},
 {1, 4, 2, 0},
 {4, 2, 9, 0}}

Output: {0, 0, 0}

Explanation: After calculating with these given coefficient the value of X, Y, Z is 0, 0, 0.

Example 2:

Input:

Arr = {{4.2, 6, 7, 10},
 {1, 2, 3, 12},

{3, 2, 2, 20}}

Output: {3, 4, -1}

Explanation: After calculating with these given coefficient the value of X, Y, Z is 3, 4, -1.

Your Task:

You don't need to read input or print anything. Complete the function **myCalculator()** which takes Arr as input parameter and returns 0 in case the system is inconsistent and 1 in case the system is consistent and has infinitely many solutions. In case the system is consistent and has a unique solution, return 3 space separated integers denoting the floor values of X, Y and Z respectively.

Expected Time Complexity: $O(1)$

Expected Auxiliary Space: $O(1)$

Constraints:

$-100 \leq \text{Arr}[i][j] \leq 100$

Nth digit of pi

Hard Accuracy: 3.23% Submissions: 3K+ Points: 8

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Calculate the **Nth** digit in the representation of Pi.

Example 1:

Input:

$N = 1$

Output:

3

Explanation:

Value of Pi is 3.14...

So, the first digit is 3.

Example 2:**Input:**

$N = 2$

Output:

1

Explanation:

Value of Pi is 3.14...

So, the second digit is 1.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **nthDigOfPi()** which takes an Integer N as input and returns the N^{th} digit of Pi.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq N \leq 10^4$

Maximum Number of Toys

Hard Accuracy: **9.59%** Submissions: **115+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given N toys in a shop .

The cost of each toy is represented by an array $A[]$. You are given Q queries, For i th query, You have a C amount of money which you can use to purchase the toys. Also there are K broken toys and you won't purchase them. The task is to calculate the maximum number of toys you can purchase using the C amount of money.

Note: 1 based indexing is used. Each query is treated independently.

Query definition: The first element represents an integer C where $C = \text{Queries}[i][0]$.

The second element represents an integer K , where $K = \text{Queries}[i][1]$.

The next K integers represent the indices of broken toys which are $\text{Queries}[i][j]$, $j > 1$

Example 1:

Input:

$N = 5$

$A[] = \{8, 6, 9, 2, 5\}$

$Q = 2$

$\text{Query}[][] = \{\{12, 2, 3, 4\}, \{30, 0\}\}$

Output:

2 5

Explanation:

Query 1: $C = 12, K = 2$,

Indices of Broken toys is $\{3,4\}$

Indices of Available toys are $\{1,2,5\}$

If we purchase the toys 2 and 5,

then $\text{cost} = A[2] + A[5] = 6 + 5 = 11$,

Therefore, We purchase the 2 toys

using 11 amount of money.

Query 2: $C = 30, K = 0$

There is no broken toy.

We can purchase all toys,

$\text{cost} = A[1] + A[2] + A[3] + A[4] + A[5] = 30$

Therefore, We purchase the 5 toys

using 30 amount of money.

Example 2:

Input:

$N = 2$

$A[] = \{3,3\}$

$Q = 1$

$\text{Query}[][] = \{\{1,0\}\}$

Output:

0

Explanation:

Query 1: $C = 1, K = 0$,

There is no broken toy.

We have not enough amount to purchase any toy.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **maximumToys()** which takes the integer N and array $A[]$, integer Q and

2D array **Queries**[][] as input parameters and returns the array of answers of each query.

Expected Time Complexity: $O(N \log M_x + Q * K * \log M_x + Q * (\log M_x)^2)$

Expected Auxiliary Space: $O(M_x)$

Where M_x is the maximum element present in the array $A[i]$.

Constraints:

$$1 \leq N \leq 10^5$$

$$1 \leq A[i] \leq 10^6$$

$$1 \leq Q \leq 10^4$$

$$1 \leq C \leq 10^9$$

$$0 \leq K \leq 10$$

$$1 \leq \text{Queries}[i][j] \leq N, j > 1$$

Kth Smallest Number in Multiplication Table

Hard Accuracy: **44.14%** Submissions: **4K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given three integers M , N and K . Consider a grid of $M * N$, where $\text{mat}[i][j] = i * j$ (1 based index). The task is to return the K^{th} smallest element in the $M * N$ multiplication table.

Example 1:

Input:

$M = 3, N = 3$

$K = 5$

Output: 3

Explanation:

1	2	3
2	4	6
3	6	9

1	2	2	3	3	4	6	6	9
---	---	---	---	---	---	---	---	---

The 5th smallest element is 3.

Example 2:

Input:

M = 2, N = 3

K = 6

Output: 6

Your Task:

You don't need to read input or print anything. Your task is to complete the function **KthSmallest()** which takes three integers as input and returns an integer as output.

Expected Time Complexity: $O(M * \log(M * N))$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq M, N \leq 3 * 10^4$

$1 \leq K \leq M * N$

Rearrange the array

Hard Accuracy: **24.41%** Submissions: **673+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an sequence from 1 to N and also given an array `arr[]` of size N. Basically the array is the permutation of 1 to N which determines that the element from i^{th} position should move to the j^{th} position. Now the task is to find the minimum number of operations such that each array element is present at its original positions.

(For more understanding please go through the examples)

Note: the answer can be large, so return the answer **modulo** 10^9+7 .

Example 1:

Input: $N = 3, arr[] = \{1, 2, 3\}$

Output: 1

Explanation:

Given special arrangement `arr[]`:

1 2 3

Given sequence

1 2 3

so 1 should go 1th position, 2 should go 2nd position and 3 should go 3rd position. So the minimum number of operation needed is 1.

Example 2:

Input: N = 5, arr[] = {2,3,1,5,4}

Output: 6

Explanation:

Given special arrangement arr[]:

2 3 1 5 4

Given sequence is:

1 2 3 4 5

so, here we explained one step,

1 should go to 2nd position, 2 should go

3rd position, 3 should go 1st position, 4

should go 5th and 5 should go 4th position.

these are the required. So after 1st operation

new sqquence will look like

3 1 2 5 4.

Here we explained the complete operations.

operations for the array

0. 1 2 3 4 5

1. 3 1 2 5 4

2. 2 3 1 4 5

3. 1 2 3 5 4

4. 3 1 2 4 5

5. 2 3 1 5 4

6. 1 2 3 4 5.

So after 6th operations the

array rearrange itself to its original positions.

Your Task:

You don't need to read or print anything. Your task is to complete the function `rearrange_array()` which take `arr[]` of size `N` as input parameter and returns an integer which denotes the minimum number of operations needed.

Expected Time Complexity: $O(N)$

Expected Space Complexity: $O(1)$

Constraints:

$1 \leq N \leq 10^5$

Concatenate two numbers

Hard Accuracy: **44.65%** Submissions: **7K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `numbers[]` of `N` positive integers and a positive integer `X`, The task is to find the number of ways that `X` can be obtained by writing pair of integers in the array `numbers[]` next to each other. In other words, find the number of ordered pairs (i, j) such that $i \neq j$ and `X` is the concatenation of `numbers[i]` and `numbers[j]`

Example 1:

Input:

`N = 4`


```
numbers[] = {1, 212, 12, 12}
```

```
X = 1212
```

Output:

```
3
```

Explanation:

We can obtain X=1212 by concatenating:

```
numbers[0] = 1 with numbers[1] = 212
```

```
numbers[2] = 12 with numbers[3] = 12
```

```
numbers[3] = 12 with numbers[2] = 12
```

Example 2:

Input:

```
N = 3
```

```
numbers[] = {11, 11, 110}
```

```
X = 11011
```

Output:

```
2
```

Explanation:

We can obtain X=11011 by concatenating:

```
numbers[2] = 110 with numbers[0] = 11
```

```
numbers[2] = 110 with numbers[1] = 11
```

Your Task:

You don't need to read input or print anything. Your task is to complete the function **countPairs()** which takes the integer **N**, the integer **X**, and the array **numbers[]** as the input parameters, and returns the number of pairs which satisfies the above condition.

Expected Time Complexity: $O(N \cdot \log_{10}(A[i]) + (\log_{10} X)^2)$

Expected Auxiliary Space: $O(N \cdot \log_{10}(A[i]))$

Constraints:

$$1 \leq N \leq 5 \cdot 10^4$$

$$1 \leq \text{numbers}[] \leq 10^9$$

$$1 \leq X \leq 10^9$$

Minimum time to fulfil all orders**Hard** Accuracy: **77.35%** Submissions: **8K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Geek is organizing a party at his house. For the party, he needs exactly N donuts for the guests. **Geek** decides to order the donuts from a nearby restaurant, which has L chefs and each chef has a rank R .

A chef with rank R can make 1 donut in the first R minutes, 1 more donut in the next $2R$ minutes, 1 more donut in $3R$ minutes, and so on.

For example, a chef with rank 2, can make one donut in 2 minutes, one more donut in the next 4 minutes, and one more in the next 6 minutes. So, it takes $2 + 4 + 6 = 12$ minutes to make 3 donuts. A chef can move on to making the next donut only after completing the previous one. All the chefs can work simultaneously. Since, it's time for the party, **Geek** wants to know the **minimum** time required in completing N donuts. Return an integer denoting the minimum time

Example 1:**Input:**

$$N = 10$$

$$L = 4$$

$$\text{rank}[] = \{1, 2, 3, 4\}$$

Output: 12**Explanation:**

Chef with rank 1, can make 4 donuts in time $1 + 2 + 3 + 4 = 10$ mins

Chef with rank 2, can make 3 donuts in time $2 + 4 + 6 = 12$ mins

Chef with rank 3, can make 2 donuts in time $3 + 6 = 9$ mins

Chef with rank 4, can make 1 donuts in time = 4 minutes

Total donuts = $4 + 3 + 2 + 1 = 10$ and total time = 12 minutes.

Example 2:

Input:

$N = 8$

$L = 8$

$\text{rank}[] = \{1, 1, 1, 1, 1, 1, 1, 1\}$

Output: 1

Explanation:

As all chefs are ranked 1, so each chef can make 1 donuts 1 min.

Total donuts = $1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 8$ and total time = 1 minute.

Your Task:

You don't need to read input or print anything. Your task is to complete the function `findMinTime()` which takes an integer N as input and an array $A[]$ of length L denoting the ranks and return an integer denoting the minimum time.

Expected Time Complexity: $O(N \cdot \log N)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq N \leq 10^3$

$1 \leq L \leq 50$

$1 \leq L[i] \leq 8$

Path of greater than equal to k length

Hard Accuracy: **36.23%** Submissions: **7K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a graph, a source vertex in the graph, and a number k, find if there is a simple path, of path length **greater than or equal to k**, (without any cycle) starting from a given source and ending at any other vertex.

Source vertex should always be 0.

Example 1:

Input:

$V = 4$, $E = 3$ and $K = 8$

$A[] = [0, 1, 5, 1, 2, 1, 2, 3, 1]$

Output: 0

Explanation:

There exists no path which has a distance of 8.

Example 2:

Input:

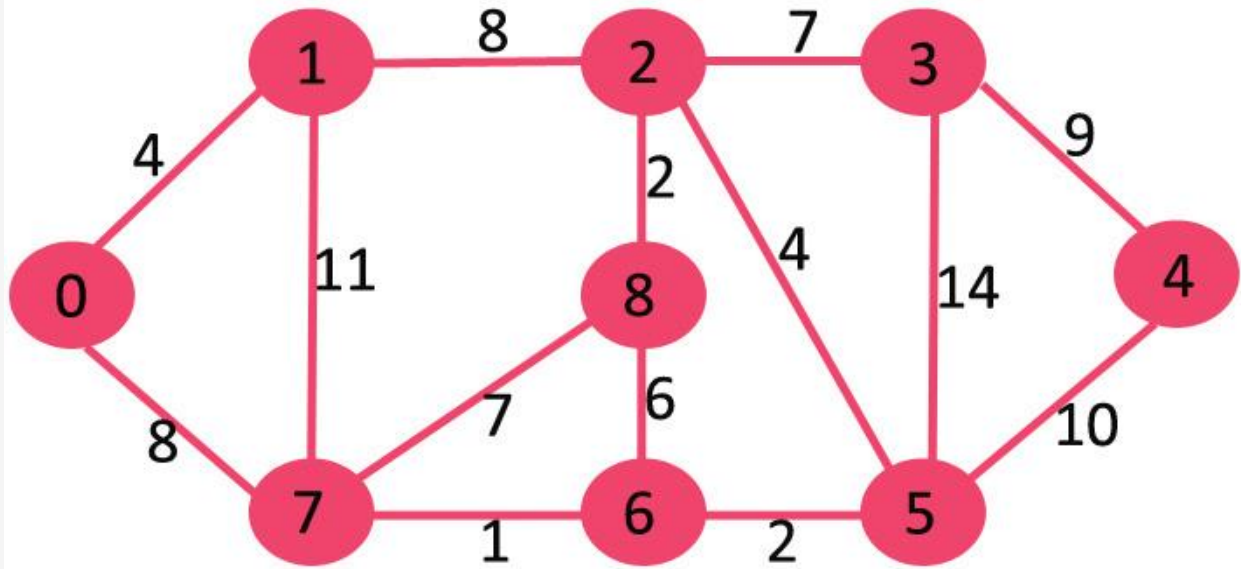
$V = 9$, $E = 14$ and $K = 60$

$A[] = [0, 1, 4, 0, 7, 8, 1, 2, 8, 1, 7, 11, 2, 3, 7, 2, 5, 4, 2, 8, 2, 3, 4, 9,$

3, 5, 14, 4, 5, 10, 5, 6, 2, 6, 7, 1, 6,
8, 6, 7, 8, 7]

Output: 0

Explanation:



Your Task:

You don't need to read input or print anything. Your task is to complete the function **pathMoreThanK()** which takes the integer V, Edges E, an integer K and Array A which is having (Source, Destination, Weight) as input parameters and returns 1 if the path of at least k distance exists, else returns 0.

Expected Time Complexity: $O(N!)$

Expected Auxiliary Space: $O(N)$

Constraints:

2 ≤ V ≤ 5

1 ≤ K ≤ 100 E ≤ 20

$1 \leq K \leq 100$

Travelling Salesman Problem

Hard Accuracy: **44.71%** Submissions: **10K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a matrix cost of size n where cost[i][j] denotes the cost of moving from city i to city j. Your task is to complete a tour from the city 0 (0 based index) to all other cities such that you visit each city atmost once and then at the end come back to city 0 in min cost.

Example 1:

Input: cost = {{0,111},{112,0}}

Output: 223

Explanation: We can visit 0->1->0 and
cost = 111 + 112.

Example 2:

Input: cost = {{0,1000,5000},{5000,0,1000},
{1000,5000,0}}

Output: 3000

Explanation: We can visit 0->1->2->0 and cost
= 1000+1000+1000 = 3000

Your Task:

You don't need to read or print anything. Your task is to complete the function `total_cost()` which takes `cost` as input parameter and returns the total cost to visit each city exactly once starting from city 0 and again comback to city 0.

Expected Time Complexity: $O(2^n * n^2)$

Expected Space Compelxity: $O(2^n * n)$

Constraints:

$1 \leq n \leq 10$

$1 \leq \text{cost}[i][j] \leq 10^3$

Optimum location of point to minimize total distance

Hard Accuracy: **49.1%** Submissions: **491** + Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a set of coordinates **points** of the form $[p, q]$ and a line **L** of the form $ax + by + c = 0$. The task is to find a point on a given line for which the sum of distances from a given set of coordinates is minimum.

Example 1:**Input:**

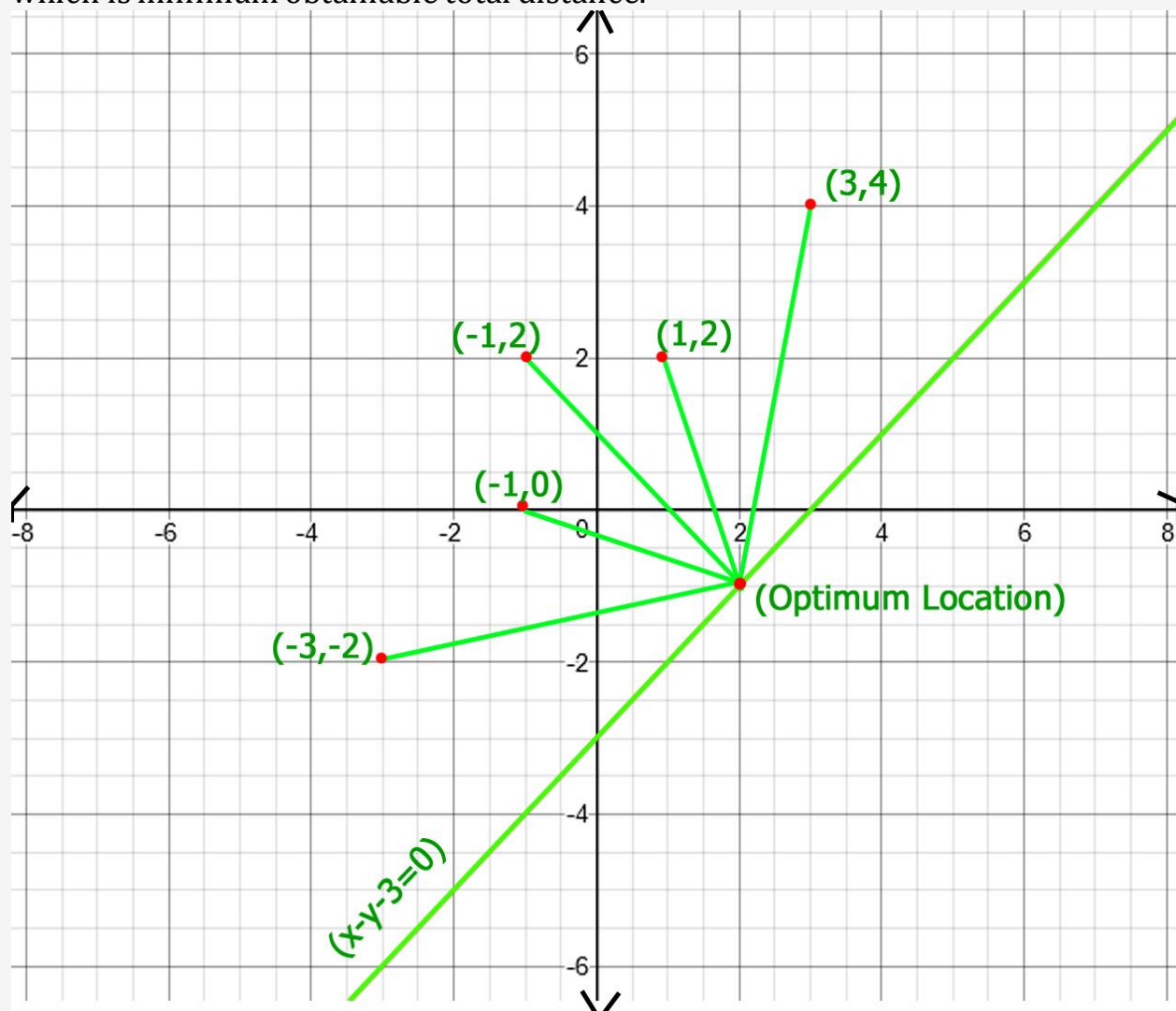
$L = \{1, -1, 3\}$

$\text{points}[] = \{-3, 2\}, \{-1, 0\},$

$\{-1, 2\}, \{1, 2\}, \{3, 4\}$

Output: 20.77

Explanation: In above figure optimum location of point of $x - y - 3 = 0$ line is $(2, -1)$, whose total distance with other points is 20.77, which is minimum obtainable total distance.



Example 2:

Input:

$L = \{2, 1, 4\}$


```
points[] = {{-1, 2}, {1, 3},{2, 4}}
```

Output: 11.20

Your Task:

You don't need to read input or print anything. Your task is to complete the function **findOptimumCost()** which takes a line **L** and coordinates and returns an double up to 2 decimal places as output.

Expected Time Complexity: $O(N \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 10^5$

$-10^3 \leq \text{point}[i] \leq 10^3$

Subset sums between the given range

Hard Accuracy: **76.17%** Submissions: **252+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array **A[]** of **N** integers and two integers **L** and **R**. The task is to return the total subsets of array **A** whose sum lies between the range **[L, R]**, inclusive.

Example 1:

Input:

A[] = {1, -2, 3}

L = -1, **R** = 2

Output: 5

Explanation:

The sum of the subsets $\{\{\}, \{1\}, \{1, -2\}, \{-2, 3\}, \{1, -2, 3\}\}$ lies between -1 and 2.

Example 2:

Input:

$A[] = \{-1, 1, 0\}$

$L = 0, R = 3$

Output: 6

Your Task:

You don't need to read input or print anything. Your task is to complete the function `countSubsets()` which takes an array of size N and integers denoting the ranges $[L, R]$.

Note : Use long long int as the answer may overflow a 32-bit integer.

Expected Time Complexity: $O(2^{N/2} * \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 20$

$-10^5 \leq A[i] \leq 10^5$

$-10^5 \leq L \leq R \leq 10^5$

Maximum difference between pair in a matrix

Hard Accuracy: **61.18%** Submissions: **201+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an $N \times N$ matrix, $\text{mat}[N][N]$ of integers. The task is to find the maximum value of $\text{mat}[c, d] - \text{mat}[a, b]$ over all choices of indexes such that both $c > a$ and $d > b$.

Example 1:

Input:

```
mat[N][N] = {{ 1, 2, -1, -4, -20 },
              { -8, -3, 4, 2, 1 },
              { 3, 8, 6, 1, 3 },
              { -4, -1, 1, 7, -6 },
              { 0, -4, 10, -5, 1 }};
```

Output: 18

Explanation: The maximum value is 18 as $\text{mat}[4][2] - \text{mat}[1][0] = 18$ has maximum difference.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **findMaxValue()** which takes a matrix **mat** and returns an integer as output.

Expected Time Complexity: $O(N^2)$

Expected Auxiliary Space: $O(N^2)$

Constraints:

$$1 \leq N \leq 10^3$$

$$-10^3 \leq \text{mat}[i][j] \leq 10^3$$

Minimum swap required to convert binary tree to binary search tree**Hard** Accuracy: **64.35%** Submissions: **2K+** Points: **8**

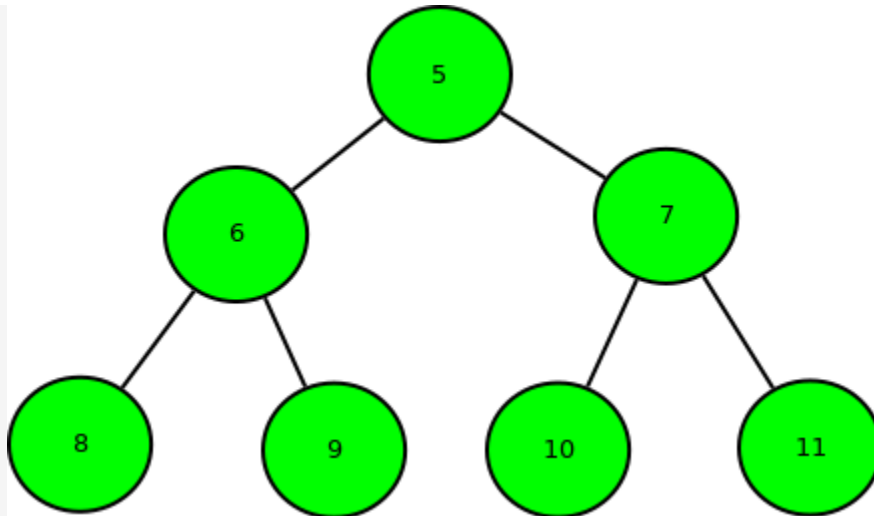
Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array $A[]$ which represents a Complete Binary Tree i.e, if index i is the parent, index $2*i + 1$ is the left child and index $2*i + 2$ is the right child.

The task is to find the minimum number of swaps required to convert it into a Binary Search Tree.

Example 1:**Input:** $A[] = \{ 5, 6, 7, 8, 9, 10, 11 \}$ **Output:** 3**Explanation:**

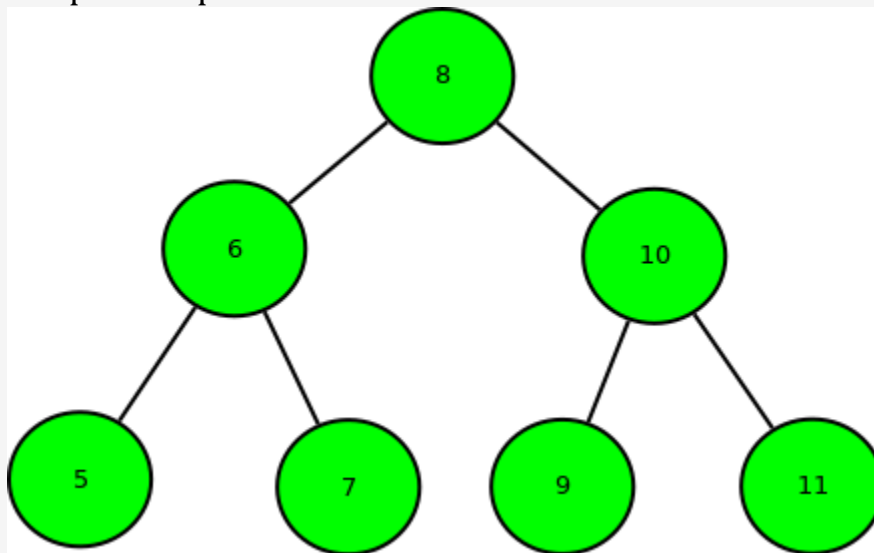
Binary tree of the given array:



Swap 1: Swap node 8 with node 5.

Swap 2: Swap node 9 with node 10.

Swap 3: Swap node 10 with node 7.



So, minimum 3 swaps are required.

Example 2:

Input:

$A[] = \{1, 2, 3\}$

Output: 1

Your Task:

You don't need to read input or print anything. Your task is to complete the function `minSwaps()` which takes an array `A[]` and returns an integer as output.

Expected Time Complexity: $O(N \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 10^5$

Largest subtree sum in a tree

Hard Accuracy: **69.8%** Submissions: **1K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a binary tree. The task is to find subtree with maximum sum in the tree and return its sum.

Example 1:**Input:**

```
    1
   / \
  2   3
 / \  / \
4 5 6 7
```

Output: 28

Explanation:

As all the tree elements are positive,

the largest subtree sum is equal to sum of all tree elements.

Example 2:

Input:

```

      1
     / \
    -2  3
   /\  /\
  4 5 -6 2

```

Output: 7

Explanation:

Subtree with largest sum is :

```

    -2
   /\
  4  5

```

Also, entire tree sum is also 7.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **findLargestSubtreeSum()** which takes the root of a binary tree and returns an integer.

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 10^5$

$-10^3 \leq \text{tree.val} \leq 10^3$

Maximum number of overlapping Intervals

Hard Accuracy: **41.52%** Submissions: **518+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given N intervals of the form $[A, B]$. The task is to return the maximum number of overlap among these intervals at any time.

Example 1:

Input:

$N = 3$

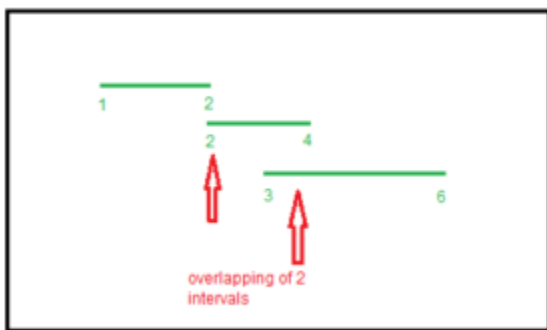
Intervals[] = {{1, 2}, {2, 4}, {3, 6}}

Output:

2

Explanation:

The maximum overlapping is 2 (between (1 2) and (2 4) or between (2 4) and (3 6))



Example 2:

Input:

$N = 4$

$\text{Intervals[]} = \{\{1, 8\}, \{2, 5\}, \{5, 6\}, \{3, 7\}\}$

Output:

4

Explanation:

The maximum overlapping is 4 (between (1, 8), (2, 5), (5, 6) and (3, 7))

Your Task:

You don't need to read input or print anything. Your task is to complete the function **overlap()** which takes a list of pairs as input and returns an integer as output.

Expected Time Complexity: $O(N * \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$2 \leq N \leq 2 * 10^4$

$1 \leq \text{intervals}[i][0] < \text{intervals}[i][1] \leq 4 * 10^3$

Longest Path in a Directed Acyclic Graph

Hard Accuracy: **40.88%** Submissions: **2K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a Weighted **D**irected **A**cyclic **G**raph (DAG) and a source vertex s in it, find the longest distances from s to all other vertices in the given graph.

Example 1:**Input:** $N=3$ $M=2$ $SRC=0$ $edges[] = \{\{0,2,1\}, \{0,1,1\}\}$ **Output:** $distance[] = \{0,1,1\}$ **Explanation:**

the shortest distance of vertex 1 from 0 is 1 and that of two is also 1.

Example 2:**Input:** $N=6$ $M=10$ $SRC=1$ $edges[] = \{\{0,1,5\}, \{0,2,3\}, \{1,3,6\}, \{1,2,2\}, \{2,4,4\}, \{2,5,2\}, \{2,3,7\}, \{3,5,1\}, \{3,4,-1\}, \{4,5,-2\}\}$ **Output:** $distance[] = \{INF, 0, 2, 9, 8, 10\}$ **Explanation:**

The vertex zero is not reachable from vertex 1 so its distance is INF, for 2 it is 2, for 3 it is 9, the same goes for 4 and 5.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **maximumDistance()** which takes the edge list **edges[]** where **edges[0]** , **edges[1]** and **edges[2]** represent **u,v** and **weight**, its size **M** and the number of nodes **N** as input parameters and returns the distance array in the distance array instead of passing **INF** you need to have **INT_MIN** driver will automatically update it to **INF**.

Expected Time Complexity: $O(V+E)$

Expected Auxiliary Space: $O(V)$

Constraints:

$$1 \leq N \leq 10^3$$

$$1 \leq M \leq N*(N-1)/2$$

$$0 \leq \text{edges}[i][0], \text{edges}[i][1]$$

$$-100 \leq \text{edges}[i][2] \leq 100.$$

Cut woods

Hard Accuracy: **26.58%** Submissions: **594+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array **A[]** of **N** integers, where **A[i]** denotes the height of **ith** tree.

Geek needs to cut down **M** meters of wood. He has a woodcutting machine that works as follows: Set the blade at a height parameter **H**, and the machine cuts off all trees having a height greater than **H**. The trees having less than

height **H** remains intact.

Since **Geek** is an environmentalist, he doesn't want to cut woods more than necessary, so he set the height **H** as high as possible.

Find the maximum height **H** of the blade that still allows to cutoff at least **M** meters of wood.

Example 1:

Input:

$A[] = \{20, 15, 10, 17\}$

$M = 7$

Output: 15

Explanation:

Set the height parameter to 15, then **Geek** can chop:

1. 1.5 metres from first tree

2. 0 metres from second tree

3. 0 metres from third tree

4. 2 metres from fourth tree

So in total **Geek** chopped $5 + 2 = 7$ metres of wood.

Example 2:

Input:

$A[] = \{4, 42, 40, 26, 46\}$

$M = 20$

Output: 36

Your Task:

You don't need to read input or print anything. Your task is to complete the

function **maxHeight()** which takes an array and an integer and returns an integer as output.

Expected Time Complexity: $O(N \cdot \log N)$

Expected Auxiliary Space: $O(1)$

Constraints:

$1 \leq N \leq 10^6$

$1 \leq A[i] \leq 10^6$

$1 \leq B \leq 2 \cdot 10^6$

Partition a set into two subsets such that the difference of subset sums is minimum (Set-2)

Hard Accuracy: **18.35%** Submissions: **289+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given a set of n integers, divide the set in two subsets, S_1 and S_2 , of $n/2$ sizes each such that the difference of the sum of two subsets is as minimum as possible. The task is to print that two subset S_1 and S_2 . Elements in S_1 and S_2 should be present in same format as given in input.

Note:

1. If n is even, then sizes of two subsets must be strictly $n/2$
2. if n is odd, then size of one subset must be $(n-1)/2$ and size of other subset must be $(n+1)/2$.
3. n is strictly greater than 1.

Example 1:

Input: `arr[] = {3, 4, 5, -3, 100, 1, 89, 54, 23, 20}`

Output: `S1 = {4, 100, 1, 23, 20}`

`S2 = {3, 5, -3, 89, 54}`

Explanation: Both output subsets are of size 5 and sum of elements in both subsets is same (148 and 148). So the minimum difference will be 0.

Example 2:

Input: `arr[] = {23, 45, -34, 12, 0, 98, -99, 4, 189, -1, 4}`

Output: `S1 = {45, -34, 12, 98, -1}`

`S2 = {23, 0, -99, 4, 189, 4}`

Explanation: The sums of elements in two subsets are 120 and 121 respectively. So the minimum difference will be 1.

Your Task:

You don't need to read input or print anything. Complete the function `minDifference()` which takes `N` and array `arr` as input parameters and returns the arrays.

Expected Time Complexity: $O(2^N)$

Expected Space Complexity: $O(2 \cdot N)$

Constraints:

$2 \leq N \leq 20$

$-10000 \leq arr[i] \leq 10000$

K Centers Problem

Hard Accuracy: 55.6% Submissions: 250+ Points: 8

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

*You have given N cities numbered from 0 to $N-1$. The distance between each pair of cities is given by $N * N$ matrix MAT where $MAT[i][j]$ denotes the distance between city i and j .*

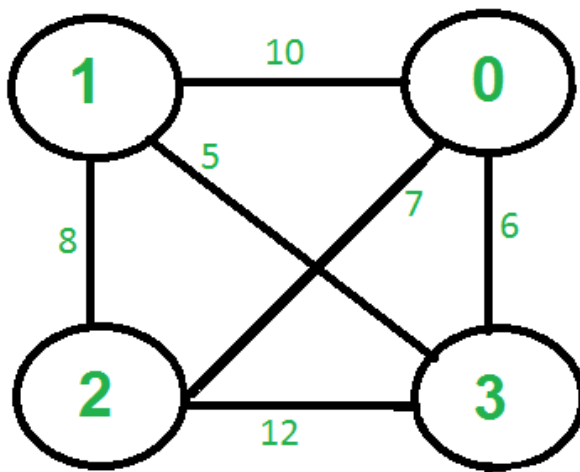
The task is to select K ($K \leq N$) ATM servers in such a way that the maximum distance of a city from the ATM Server is minimized.

Example 1:

Input: $N = 4, K = 2,$
 $MAT[][] = \{\{0, 10, 7, 6\},$
 $\{10, 0, 8, 5\},$
 $\{7, 8, 0, 12\},$
 $\{6, 5, 12, 0\}\}$

Output: 6

Explanation:



$k = 2$

The two ATMs should be placed in cities 2 and 3. The maximum distance of a city from an ATM becomes 6 in this optimal placement (We can not get the maximum distance less than 7)

Your Task:

You don't need to read or print anything. Your task is to complete the function **selectKcities()** which takes N, K and MAT[][] as input parameter and returns an integer, indicating the maximum distance of a city from the ATM Server, which is minimized.

Expected Time Complexity: $O(N * K * (2^N))$

Expected Space Complexity: $O(K)$

Constraints:

$1 \leq K \leq N \leq 15$

$1 \leq \text{MAT}[i][j] \leq 10^9$

Maximum Triple Product

Hard Accuracy: **29.77%** Submissions: **581** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given an array **arr** of **N** integers. You need to find the **maximum** score of an array.

The score of an array is calculated as follows.

- Choose an integer **i**, $1 \leq i \leq \text{size of current array}$ and add $a[i-1] * a[i] * a[i+1]$ to the score.
- Delete **a[i]**, on deleting **a[i]**, **i-1** and **i+1** become adjacent.
i.e **i+1** becomes **i** and size of an array **shrinks**.
- Repeat the process till the size of an array becomes **0**.

Note: Assume an extra 1 at each boundary.

Example 1:

Input:

$N = 2$

$\text{arr} = \{ 5, 10 \}$

Output:

60

Explanation:

First choose $i=1$, $\text{Score} = 1*5*10$

Then choose $i=1$, $\text{Score} += 1*10*1$,

Total = 60

Example 2:

Input:

$N = 5$

$\text{arr} = \{ 1, 2, 3, 4, 5 \}$

Output:

110

Your Task:

You do not need to read input or print anything. Your task is to complete the function **maxProductSum()** which takes the value **N** and the array as input parameters and returns **maximum score of an array**.

Expected Time Complexity: $O(N^3)$

Expected Auxiliary Space: $O(N^2)$

Constraints:

$$1 \leq N \leq 100$$

$$1 \leq \text{arr}[i] \leq 100000$$

Maximum Connected group**Hard** Accuracy: **43.9%** Submissions: **1K+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given an $n \times n$ binary grid. A grid is said to be binary if every value in grid is either **1** or **0**.

You can change at **most one** cell in grid from **0** to **1**.

You need to find the largest group of connected **1**'s.

Two cells are said to be connected if both are **adjacent** to each other and both have same value.

Example 1**Input:**

2

1 1

0 1

Output:

4

Explanation:

By changing cell (2,1) ,we can obtain a connected group of 4 1's

1 1

1 1

Example 2

Input:

3

1 0 1

1 0 1

1 0 1

Output:

7

Explanation:

By changing cell (3,2) ,we can obtain a connected group of 7 1's

1 0 1

1 0 1

1 1 1

Your Task:

You don't need to read or print anything. Your task is to complete the function **MaxConnection()** which takes a matrix `grid[][]` denoting the grid and return the maximum group of connected group of 1s.

Constraints:

$1 \leq n \leq 500$

$0 \leq \text{grid}[i][j] \leq 1$

Minimize Max Distance to Gas Station

Hard Accuracy: **30.52%** Submissions: **288+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

We have an horizontal number line. On that number line, we have gas **stations** at positions `stations[0]`, `stations[1]`, ..., `stations[N-1]`, where **N** = size of the stations array. Now, we add **K** more gas stations so that **D**, the maximum distance between adjacent gas stations, is minimized. We have to find the smallest possible value of **D**. Find the answer **exactly** to 2 decimal places.

Example 1:

Input:

`N = 10`

`stations = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`

`K = 9`

Output: 0.50

Explanation: Each of the 9 stations can be added mid way between all the existing adjacent stations.

Example 2:

Input:

`N = 10`

`stations = [3,6,12,19,33,44,67,72,89,95]`

`K = 2`

Output: 14.00

Explanation: Construction of gas stations at 86 locations

Your Task:

You don't need to read input or print anything. Your task is to complete the function **findSmallestMaxDist()** which takes a list of stations and integer K as inputs and returns the smallest possible value of D. Find the answer **exactly** to 2 decimal places.

Expected Time Complexity: $O(N \cdot \log K)$

Expected Auxiliary Space: $O(1)$

Constraint:

$10 \leq N \leq 5000$

$0 \leq \text{stations}[i] \leq 10^9$

$0 \leq K \leq 10^5$

stations is sorted in a **strictly increasing** order

LFU cache

Hard Accuracy: **48.29%** Submissions: **29+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Design and implement a data structure for a Least Frequently Used (LFU) cache.

Implement the given LFUCache Class with the following description.

LFUCache(int capacity): This Initializes the object with the capacity of the data structure.

int get(int key): Returns the value of the given key if it exists or else returns -1.

void put(int key, int value): If the key is already present, then update its value and if the key is not present, then it should be inserted. If the data structure has reached its maximum capacity, then the least frequently used (LFU) key should be invalidated and removed. If during removal, more than one key has same frequency, then the Least Recently Used (LRU) key amongst them should be removed.

Example:

Input:

Capacity: 3

put(5,7)put(4,6)put(3,5)put(2,4)put(1,3)

get(1)get(2)get(3)get(4)get(5)

Output:

3

4

5

-1

-1

Explanation:

When put(2,4) is executed, (5,7) gets invalidated. Similarly when put(1,3) is executed, (4,6) gets invalidated. Therefore only the values for key 1,2,3 are present in the cache after all the put operations.

Your Task:

Implement the given functions, void put(int key, int value) and int get(int key).

The functions get and put must each run in $O(1)$ average time complexity.

Expected Time Complexity: $O(N)$.

Expected Auxiliary Space: $O(1)$.

Constraints:

$0 \leq \text{capacity} \leq 10^4$

$0 \leq \text{key} \leq 10^5$

$0 \leq \text{value} \leq 10^6$

At most $2 * 10^5$ calls will be made to get and put.

Count Reverse Pairs

Hard Accuracy: **36.84%** Submissions: **220+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

You are given an array of N integers **arr**, find the count of reverse pairs.

A pair of indices (i, j) is said to be a **reverse pair** if both the following conditions are met:

- $0 \leq i < j < N$
- $\text{arr}[i] > 2 * \text{arr}[j]$

Example 1:

Input:

$N = 6$

$\text{arr} = [3, 2, 4, 5, 1, 20]$

Output:

3

Explanation:

The Reverse pairs are

- (i) (0, 4), arr[0] = 3, arr[4] = 1, $3 > 2(1)$
- (ii) (2, 4), arr[2] = 4, arr[4] = 1, $4 > 2(1)$
- (iii) (3, 4), arr[3] = 5, arr[4] = 1, $5 > 2(1)$

Example 2:

Input:

N = 5

arr= [2, 4, 3, 5, 1]

Output:

3

Explanation:

- (i) (1, 4), arr[1] = 4, arr[4] = 1, $4 > 2 * 1$
- (ii) (2, 4), arr[2] = 3, arr[4] = 1, $3 > 2 * 1$
- (iii) (3, 4), arr[3] = 5, arr[4] = 1, $5 > 2 * 1$

Your Task:

Complete the function **countRevPairs()**, which takes integer a list of N integers as input and returns the count of Reverse Pairs.

Expected Time Complexity: $O(N \log N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 50000$ $1 \leq \text{arr}[i] \leq 10^9$ **Minimum Window Subsequence****Hard** Accuracy: **33.98%** Submissions: **106+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given strings **str1** and **str2**, find the minimum (contiguous) substring **W** of **str1**, so that **str2** is a subsequence of **W**.

If there is no such window in **str1** that covers all characters in **str2**, return the empty string "". If there are multiple such minimum-length windows, return the one with the left-most starting index.

Example 1:**Input:**

str1: geeksforgeeks

str2: eksrg

Output:

eksforg

Explanation:

Eksforg satisfies all required conditions. str2 is its subsequence and it is longest and leftmost among all possible valid substrings of str1.

Example 2:**Input:**

str1: abcdebdde

str2: bde

Output:

bcde

Explanation:

"bcde" is the answer and "deb" is not a smaller window because the elements of T in the window must occur in order.

Your Task:

Complete the function `string minWindow(string str1, string str2)`, which takes two strings as input and returns the required valid string as output.

Expected Time Complexity: $O(N^2)$.

Expected Auxiliary Space: $O(N^2)$.

Constraints:

$0 \leq \text{str1} \leq 1000$

$0 \leq \text{str2} \leq 100$

Maximum XOR With an Element From Array

Hard Accuracy: **30.62%** Submissions: **49+** Points: **8**

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

Given an array `arr` of size `N` consisting of non-negative integers. You are also given `Q` queries represented by 2D integer array `queries`, where `queries[i] = [xi, mi]`.

The answer to the i^{th} query is the **maximum** bitwise XOR value of x_i and any

element of arr that does not exceed m_i . In other words, the answer is $\max(\text{arr}[j] \text{ XOR } x_i)$ for all j such that $\text{arr}[j] \leq m_i$. If all elements in nums are larger than m_i , then answer is -1.

Return an integer array **answer** where **answer[i]** is the answer to the i^{th} query.

Example 1:

Input:

N = 5

Q = 3

arr [] = {0, 1, 2, 3, 4}

queries [][] = {{3, 1}, {1, 3}, {5, 6}}

Output: {3, 3, 7}

Explanation:

1. 0 and 1 are the only two integers not greater than 1. $0 \text{ XOR } 3 = 3$ and $1 \text{ XOR } 3 = 2$. The larger of the two is 3.
2. $1 \text{ XOR } 2 = 3$.
3. $5 \text{ XOR } 2 = 7$.

Example 2:

Input:

N = 6

Q = 3

arr [] = {5, 2, 4, 6, 6, 3}

queries [][] = {{12, 4}, {8, 1}, {6, 3}}

Output: {15, -1, 5}

Your Task:

You don't need to read input or print anything. Your task is to complete the function **maximumXor()** which takes the interger N, integer Q, integer array **arr** [

] and 2D integer array **queries** [][] as parameters and returns integer array where, the i^{th} element is the answer to the i^{th} query.

Expected Time Complexity: $O(\max(Q \log Q, N \log N))$

Expected Auxiliary Space: $O(Q + N)$

Constraints:

$1 \leq N, Q \leq 10^5$

$0 \leq \text{arr}_i, x_i, m_i \leq 10^9$

Minimum Number Of Sprinkler

Hard Accuracy: 0.0% Submissions: 0 Points: 8

Bag Offers from Top Product Companies. Explore Exclusive Problems Now!

There is a one-dimensional garden of length N . In each position of the N length garden, a sprinkler has been installed. Given an [array](#) $a[]$ such that $a[i]$ describes the coverage limit of the i^{th} sprinkler. A sprinkler can cover the range from the position $\max(i - a[i], 1)$ to $\min(i + a[i], N)$. In beginning, all the sprinklers are switched off.

The task is to find the minimum number of sprinklers needed to be activated such that the whole N -length garden can be covered by water.

Note: Array is 1-based indexed.

Example 1:

Input: $a[] = \{1, 2, 1\}$

Output: 1

Explanation:

For position 1: $a[1] = 1$, range = 1 to 2

For position 2: $a[2] = 2$, range = 1 to 3

For position 3: $a[3] = 1$, range = 2 to 3

Therefore, the fountain at position $a[2]$ covers the whole garden. Therefore, the required output is 1.

Example 2:

Input: $a[] = \{2, 1, 1, 2, 1\}$

Output: 2

Explanation:

For position 1: $a[1] = 2$, range = 1 to 3

For position 2: $a[2] = 1$, range = 1 to 3

For position 3: $a[3] = 1$, range = 2 to 4

For position 3: $a[4] = 2$, range = 2 to 5

For position 3: $a[5] = 1$, range = 4 to 5

Therefore, the fountain at position $a[1]$ and $a[4]$ covers the whole garden.

Therefore, the required output is 2. Also possible answer is $a[2]$ and $a[4]$.

Your Task:

Your task is to complete the function **minSprinkler()** which takes an integer array **a** and an integer **N** as the input parameters and returns an integer denoting the minimum number of sprinkler needed to be activated such that the whole **N**-length garden can be covered by water.

Constraints:

- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq arr[i] \leq 10^9$