

## 1..... Maximum AND Value

Given an array **arr[]** of **N** positive elements. The task is to find the Maximum **AND** Value generated by any pair(**arr<sub>i</sub>**, **arr<sub>j</sub>**) from the array such that **i != j**.

**Note:** AND is bitwise '&' operator.

**Input:**

N = 4

arr[] = {4, 8, 12, 16}

**Output:** 8

**Explanation:**

Pair (8,12) has the *Maximum AND Value* 8.

## Example 2:

**Input:**

N = 4

arr[] = {4, 8, 16, 2}

**Output:** 0

**Explanation:** Any two pairs of the array has *Maximum AND Value* 0.

```
public static int checkBits(int[] ar, int mask)
{
    int count = 0;

    for(int i=0; i<ar.length; i++)
    {
        if( (mask & ar[i]) == mask)
        {
            count++;
        }
    }
    return count;
}

public static int maxAndValue(int[] ar)
{
    int mask = 0;
    int res = 0;
    |
```

```

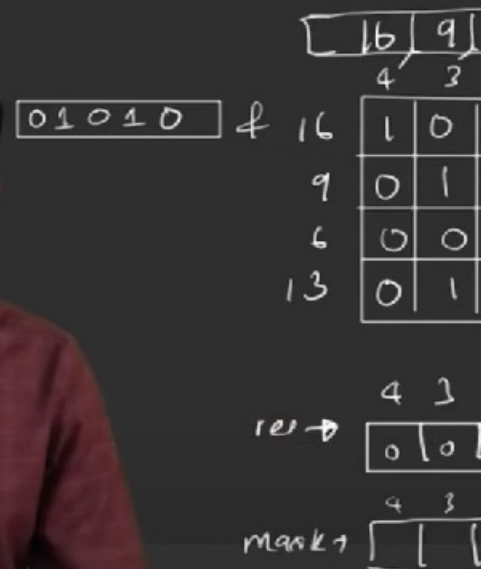
        count++;
    }
    return count;
}

public static int maxAndValue(int[] ar)
{
    int mask = 0;
    int res = 0;

    for(int i = 31; i >= 0 ; i++)
    {
        mask = (1<<i) | res;
        int count = checkBits(ar, mask);

        if(count >= 2)
        {
            res = res | mask;
        }
    }
    return res;
}

```



2..... Count total set bits

You are given a number N. Find the total count of set bits for all numbers from 1 to N(both inclusive).

**Input:** N = 4

**Output:** 5

**Explanation:**

For numbers from 1 to 4.

For 1: 0 0 1 = 1 set bits

For 2: 0 1 0 = 1 set bits

For 3: 0 1 1 = 2 set bits

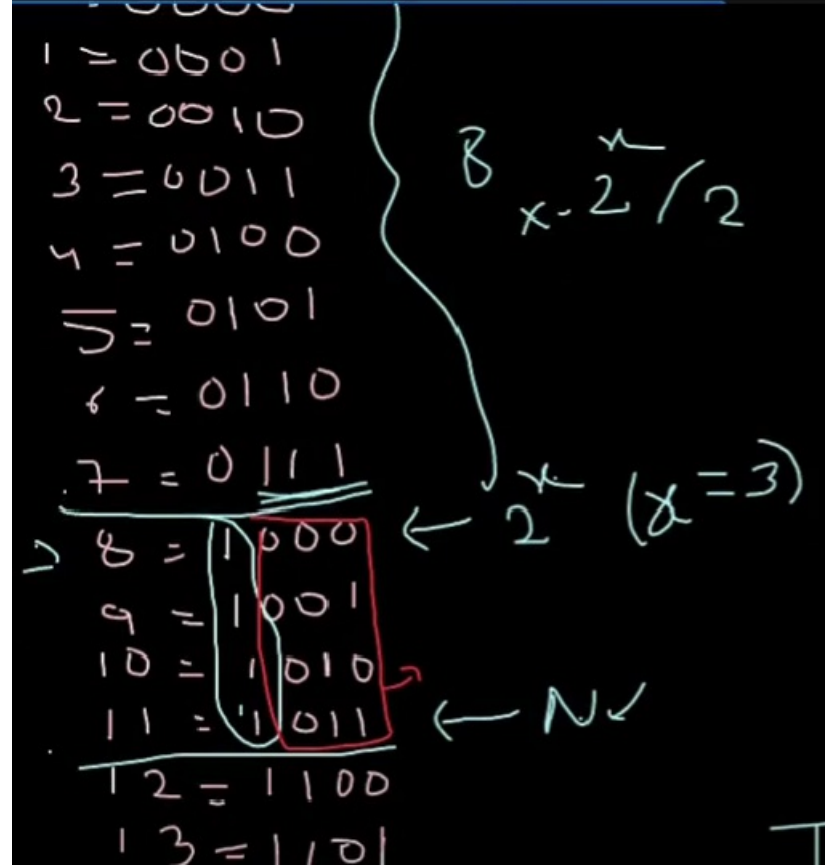
For 4: 1 0 0 = 1 set bits

Therefore, the total set bits is 5.

**Input:** N = 17

**Output:** 35

**Explanation:** From numbers 1 to 17(both inclusive), the total number of set bits is 35.



$$T_{\text{tot}} = \frac{x \cdot 2^x}{2} + N - 2^x + 1 + T_{\text{tot}}(x)$$

<https://www.youtube.com/watch?v=feaz6EsLZ6A>

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-BitMagic/problem/count-total-set-bits-1587115620>

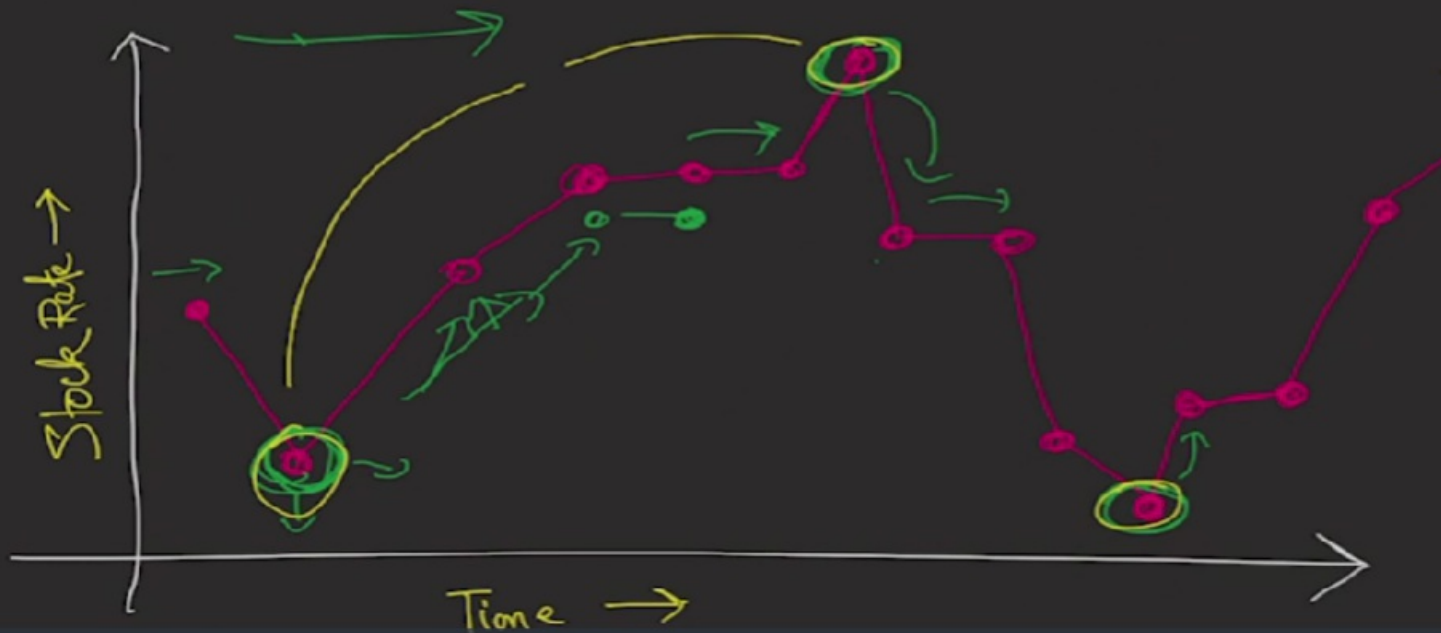
return res+countSetBits(n-pow(2,x));

Arrays

1..... Stock buy and sell

Find all the segments of days on which you buy and sell the stock so that in between those days for which profit can be generated.

# Stock Buy and Sell



Local minima par buy and maxima par sell until reaches size

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Arrays/problem/stock-buy-and-sell-1587115621>

## 2 ..... Minimum Multiplications to reach End

Given start, end and an array arr of n numbers. At each step, start is multiplied with any number in the array and then mod operation with 100000 is done to get the new start.

Your task is to find the minimum steps in which end can be achieved starting from start. If it is not possible to reach end, then return -1.

### Input:

arr[] = {2, 5, 7}

start = 3, end = 30

### Output:

2

### Explanation:

Step 1:  $3 \times 2 = 6 \ \% \ 100000 = 6$

Step 2:  $6 \times 5 = 30 \ \% \ 100000 = 30$

### Input:

arr[] = {3, 4, 65}

start = 7, end = 66175

### Output:

4

### Explanation:

Step 1:  $7 \times 3 = 21 \ \% \ 100000 = 21$

Step 2:  $21 \times 3 = 63 \ \% \ 100000 = 63$

Step 3:  $63 \times 65 = 4095 \ \% \ 100000 = 4095$

Step 4:  $4095 \times 65 = 266175 \ \% \ 100000 = 66175$

Approach: Initially start ka level 0 hai phir usse array ke sabhi element se multiply karke check karege if it reaches end or not. **BFS Lagega**. Level increment karna mushkil hai isliye queue me leve aur element ka pair push kiya aur uske child ko push karte waqt level+1 kiya

Bonus : Check if start==end then return 0

<https://practice.geeksforgeeks.org/problems/minimum-multiplications-to-reach-end/1>

## 3..... Count only Repeated

Given an array arr[] of N positive integers, where elements are consecutive (sorted). Also, there is a single element which is repeating X (any variable) number of times. Now, the task is to find the element which is repeated and number of times it is repeated.

Note: If there's no repeating element, Return {-1,-1}.

<b>Input:</b> N = 5 arr[] = {1,2,3,3,4}	<b>Input:</b> N = 5 arr[] = {2,3,4,5,5}
<b>Output:</b> 3 2	<b>Output:</b> 5 2
<b>Explanation:</b> In the given array, 3 is occuring two times.	<b>Explanation:</b> In the given array, 5 is occuring two times.

Approach: Binary Search laga to find left ya right koi mid ke barabar ho, iss se repeated element mil jayega. No. of repeated element ke liye:  $n - (arr[n-1] - arr[0])$  . max difference do different consecutive me 1 ka hi hai.

Base Case agar koi repeated no. nhi hai  $if(arr[0]+n-1==arr[n-1])$   
 return make\_pair(-1,-1);

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Searching/problem/count-only-repeated2047>

#### 4..... Maximum K valid Sum Combination

Given two integer array A and B of size N each. A sum combination is made by adding one element from array A and another element of array B. Return the maximum K valid sum combinations from all the possible sum combinations. Note : Output array must be sorted in non-increasing order.

<b>Input:</b> N = 2 K = 2 A [ ] = {3, 2} B [ ] = {1, 4} <b>Output:</b> {7, 6} <b>Explanation:</b> 7 -> (A : 3) + (B : 4) 6 -> (A : 2) + (B : 4)	<b>Input:</b> N = 4 K = 3 A [ ] = {1, 4, 2, 3} B [ ] = {2, 5, 1, 6} <b>Output:</b> {10, 9, 9} <b>Explanation:</b> 10 -> (A : 4) + (B : 6) 9 -> (A : 4) + (B : 5) 9 -> (A : 3) + (B : 6)
---	--

Approach : Yaha Naïve is All Combination form kar then sort then last k is answer

Optimal Approach : Since yaha **first k elements** nikalne hai with maximum sum then we have to go with **priority queue**.

Pehle Sort both then all unique points

<https://practice.geeksforgeeks.org/problems/maximum-sum-combination/1>

→ Sort.

→ Push —, —, —, —

→  $(x, y)$   
 $(x, y-1)$   
 $(x-1, y)$

5..... Find the closest pair from two arrays :

Given two **sorted** arrays **arr** and **brr** and a number **x**, find the pair whose **sum** is closest to **x** and the pair has an element from **each** array. In the case of multiple closest pairs return any one of them.

Note: Can return the two numbers in any manner.

**Input :** N = 4, M = 4

arr[ ] = {1, 4, 5, 7}

brr[ ] = {10, 20, 30, 40}

X = 32

**Output :**

1, 30

**Explanation:**

The closest pair whose sum is closest to 32 is {1, 30} = 31.

**Approach : Since** dono array sorted hai to two pointer approach . Ek pointer ek ke starting me and ek dusre ke last element pe point karwao. Agar Absolute of difference choota hoga to result ko do baar pop karke result me dono point daal de

Agar dono pointers ka z se difference 0 ho to break the loop warna check ki dono pointer ka sum x se choota hai ya bada **Choota ho to** Aage wala pointer increase aur bada ho to peche wala pointer decrease

<https://practice.geeksforgeeks.org/problems/find-the-closest-pair-from-two-arrays4215/1>

6..... Left most and right most index:

Given a sorted array with possibly duplicate elements. The task is to find indexes of first and last occurrences of an element **X** in the given array.

**Note:** If the element is not present in the array **return {-1,-1} as pair**.

**Input:**

$N = 9$

$v[] = \{1, 3, 5, 5, 5, 5, 67, 123, 125\}$

$X = 5$

**Output:**

2 5

**Explanation:**

Index of first occurrence of 5 is 2

and index of last occurrence of 5 is 5.

Approach : Pehle element find kar by binary search then if not found -1 -1 return else left and right jao till not equal to x or till extreme

<https://practice.geeksforgeeks.org/problems/find-first-and-last-occurrence-of-x0849/1>

Searching :

The minimum number of comparisons required to determine if an integer appears more than  $n/2$  integers is

☐  $\Theta(n)$

☒  $\Theta(\log n)$

☐  $\Theta(n \cdot \log n)$

☐  $\Theta(1)$

A matrix  $M(3 \times 5)$  is to be stored in column-major order in a computer. If the base address of the matrix is 1024 and each element is 4 bytes, then what will be the address of  $M(2, 3)$ ?

- (A) 1028
- (B) 1188
- (C) 1204
- (D) 1220



Let A be a square matrix of size  $n \times n$ . The maximum number of elements that can be nonzero in A without overflow is  $2^p$ . What is the minimum possible value of p?

☐  $\log_2(n)$

☐  $\log_2(n/2)$

☐  $\log_2(n-1)$

☐  $\log_2(n+1)$

Strings :

1....Count number of substrings : Given a string of lowercase alphabets, count all possible substrings (not necessarily distinct) that have exactly k distinct characters.

Linked List :

1..... Delete nodes having greater value on right :

Given a singly linked list, remove all the nodes in the list which have any node on their right whose value is greater. (Not just immediate Right , but entire List on the Right)

**Input:**

LinkedList = 12->15->10->11->5->6->2->3

**Output:** 15 11 6 3

**Explanation:** Since, 12, 10, 5 and 2 are the elements which have greater elements on the following nodes. So, after deleting them, the linked list would like be 15, 11, 6, 3.

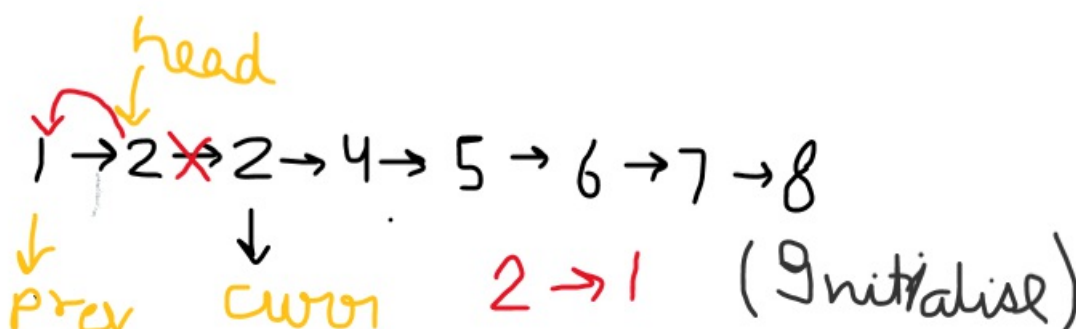
Approach :

Reverse kar ke delete karna then again reverse

<https://practice.geeksforgeeks.org/problems/delete-nodes-having-greater-value-on-right/1>

In While loop

Iteration 1:





Matrix

1... Boolean Matrix :

Given a boolean matrix of size  $R \times C$  where each cell contains either 0 or 1, modify it such that if a matrix cell  $matrix[i][j]$  is 1 then all the cells in its  $i^{th}$  row and  $j^{th}$  column will become 1.

Approach: Jaha bhi 1 hai uska row and column set me store kara le then set me ja and uss row aur column ko 1 kar de

<https://practice.geeksforgeeks.org/problems/boolean-matrix-problem-1587115620/1>

2. Determinant of matrix :

Given a square matrix of size  $n \times n$ . The task is to find the **determinant** of this matrix.

Approach

base cases  $n=1$  and  $n=2$  go with formula and kick is minor matrix ek ag function se banya hai.

```
vector<vector<int>> submat(vector<vector<int>> m,int j, int n){
    vector<vector<int>> res;
    for(int i=1;i<n;i++){
        vector<int> t;
        for(int k=0;k<n;k++){
            if(k==j)
                continue;
            t.push_back(m[i][k]);
        }
        res.push_back(t);
    }
    return res;
}

public:
//Function for finding determinant of matrix.
int determinantOfMatrix(vector<vector<int>> m, int n)
{
    // code here
    if(n==1) return m[0][0];
    if(n==2) return ((m[0][0]*m[1][1])-(m[1][0]*m[0][1]));

    int det=0;
    for(int j=0;j<n;j++){
        det+=pow(-1,j+0)*m[0][j]*determinantOfMatrix(submat(m,j,n),n-1);
    }
    return det;
}
```

Trees

1..... Leftmost and rightmost nodes of binary tree

Given a Binary Tree of size  $N$ , Print the corner nodes ie- the node at the leftmost and rightmost of each level.

**Input:**

```
      10
     /  \
    20   30
   /  \
  40  60
```

**Output:** 10 20 30 40 60

Approach : Go with level order traversal using queue but each level ko alag kar using NULL

10 NULL

NULL 20 30

20 30 NULL

While curr!=NULL vector me front ka data push kar queue me child push kar and queue se pop kar

Vector me level elemets and then first and last according to size

<https://practice.geeksforgeeks.org/problems/leftmost-and-rightmost-nodes-of-binary-tree/1>

2..... Leaf under budget

Given a binary tree and a **budget**. Assume you are at the root of the tree(**level 1**), you need to maximise the count of leaf nodes you can visit in your budget if the **cost of visiting** a leaf node is equal to the **level of that leaf node**.

**Input:**



and budget = 8

**Output: 2**

**Explanation:**

Cost For visiting Leaf Node 3: 3

Cost For visiting Leaf Node 4: 4

Cost For visiting Leaf Node 6: 3

In budget 8 one can visit Max 2 Leaf Nodes.

**Input:**



and budget = 5

**Output: 1**

**Explanation:** We can only visit either node 4 or 5.

Isme level order traverse kar aur agar leaf ode ho mtlb left aur right dono null ho to cost(i.e., level ko ek vector me store kar le ) aur har level me level++ karte rahio from 0 to h

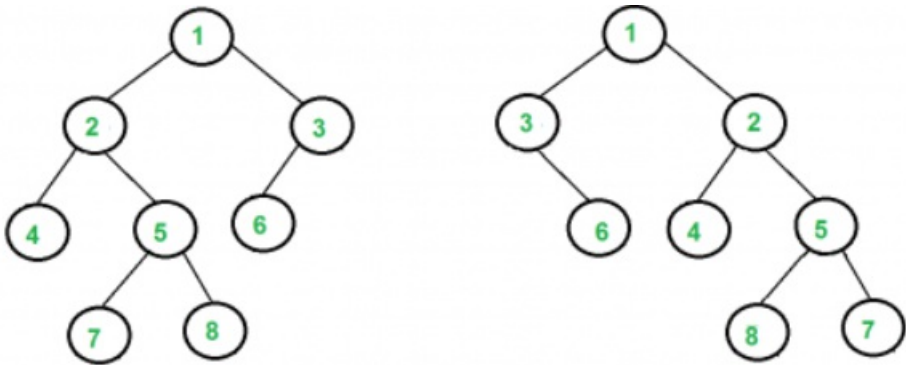
Cost vector ban jaayega to budget ke according maximum nodes select kar lio

<https://practice.geeksforgeeks.org/problems/leaf-under-budget/1>

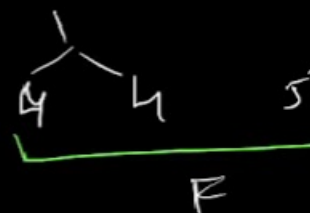
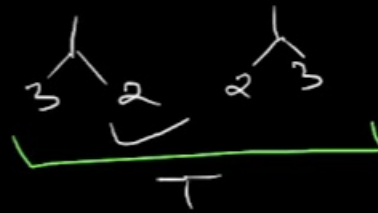
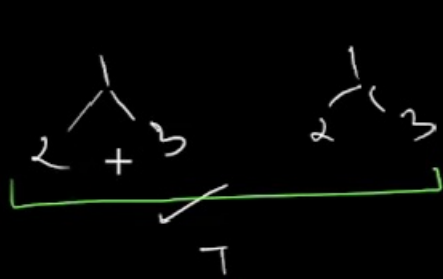
3..... Check if two trees are isomorphic

Two trees are called isomorphic if one can be obtained from another by a series of flips, i.e. by swapping left and right children of several nodes. Any number of nodes at any level can have their children swapped. Two empty trees are isomorphic.

For example, the following two trees are isomorphic with the following sub-trees flipped: 2 and 3, NULL and 6, 7 and 8.

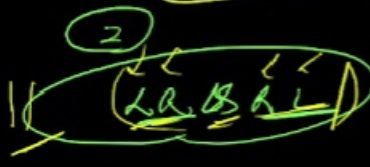
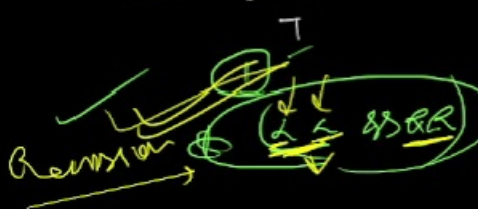
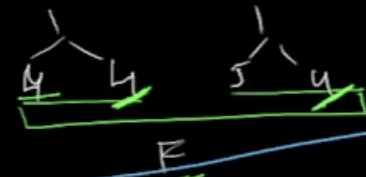


# Check if Tree is Isomorphic



# Check if Tree is Isomorphic

high POTD  
D-310/363

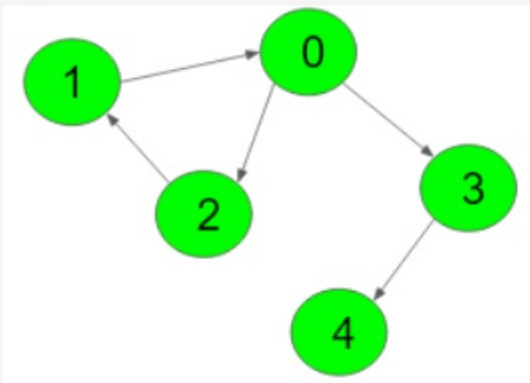


if (r1 == null && r2 == null)  
return T;

if (r1 == null || r2 == null)  
return F;

else if (r1.left == r2.left && r1.right == r2.right) || (r1.left == r2.right && r1.right == r2.left) return T;  
else return F;

Input:



**Output:** 0

**Explanation:** According to the given edges, all nodes can be reached from nodes 0, 1 and 2. But, since 0 is minimum among 0, 1 and 2, so 0 is the output.

Approach :

Mera to queue based approach tha but vo tle de gaya complexity( $V*(V+E)$ )

Usme har node ko check kiya tha whether uss se saare node visit ho rahe hai ya nahi with help of visit vector , count variable ; BFS ya DFS koi bhi approach dfs kiya tha maine

Optimized Solution

Go with dfs only but 1 baar iterate kar bas

Visit ko 1 kar by dfs like  $i=0$  iteration me 0,2,3,4 index 1 ho jayenge iss stage pe mother candidate 0 hoga

But next iteration  $i=1$  visit nhi hai

To  $i=1$  par dfs chalega jab and phir for loop khatam ho jayega **with each vertex 1 baar chalega**

**Ab visit ko 0 kar de aur mother candidate ke saath dfs call kar**

**Agar sab 1 ho gaye to mother hai warna nhi hai in that case return -1**

<https://practice.geeksforgeeks.org/problems/mother-vertex/1>

Sorting

1..... Merge three sorted arrays

**Input:**

$N = 4, A[] = [1\ 2\ 3\ 4]$

$M = 5, B[] = [1\ 2\ 3\ 4\ 5]$

$P = 6, C[] = [1\ 2\ 3\ 4\ 5\ 6]$

**Output:** 1 1 1 2 2 2 3 3 3 4 4 4 5 5 6

**Explanation:** Merging these three sorted arrays, we have:

1 1 1 2 2 2 3 3 3 4 4 4 5 5 6.

Approach : pehle 2 array merge kar then resultant array/vector ko 3<sup>rd</sup> vector ke saath merge kar de

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Sorting/problem/merge-three-sorted-arrays-1587115620>

2.. Find Triplets with sum 0 or X

Given an array `arr[]` of  $n$  integers. Check whether it contains a triplet that sums up to zero.

Note: Return 1, if there is at least one triplet following the condition else return 0.

Follow the steps below to implement the idea:

- Sort the array in ascending order.
- Traverse the array from start to end.
- For every index  $i$ , create two variables  $l = i + 1$  and  $r = n - 1$
- Run a loop until  $l$  is less than  $r$  if the sum of `array[i]`, `array[l]` and `array[r]` is equal to zero then print the triplet and break the loop
- If the sum is less than zero then increment the value of  $l$ , by increasing the value of  $l$  the sum will increase as the array is sorted, so `array[l+1] > array[l]`
- If the sum is greater than zero then decrement the value of  $r$ , by decreasing the value of  $r$  the sum will decrease as the array is sorted, so `array[r-1] < array[r]`.

<https://www.geeksforgeeks.org/batch/dsa-4/track/DSASP-Sorting/problem/find-triplets-with-zero-sum>

3.... Sort if triple less than a than a to b then greater than b

<https://www.geeksforgeeks.org/batch/dsa-4/track/DSASP-Sorting/problem/three-way-partitioning>

```
void threeWayPartition(vector<int>& arr,int a, int b)
{
    // code here
    int l=0;
    int n=arr.size();
    int r=n-1;
    for(int i=0;i<=r;i++){
        if(arr[i]<a){
            swap(arr[i],arr[l]);
            l++;
        }
        else if(arr[i]>b)
        {
            swap(arr[i],arr[r]);
            r--;
            i--;    // jab i swap hoga to i ko bhi check
            // karna hoga so i-- so i++ nullify ho jaye
        }
    }
}
```

4. Kth smallest : Sort karke return `arr[k]` return kar de ya hir use priority queue

5.. Minimum difference in array : Sort karke minimize the difference between `arr[i]` and `arr[i+1]`

6. Chocolate Distribution Problem : M size ka minimum difference after sorting

Dynamic Programming

1..... Stickler Thief: Stickler the thief wants to loot money from a society having  $n$  houses in a single line. He is a weird person and follows a certain rule when looting the houses. According to the rule, he will never loot two consecutive houses. At the same time, he wants to maximize the amount he loots. The thief knows which house has what amount of money but is unable to come up with an optimal looting strategy. He asks for your help to find the maximum money he can get if he strictly follows the rule.  $i$ th house has `a[i]` amount of money present in it.

**Input:**

`n = 5`

`a[] = {6,5,5,7,4}`

**Output:**

15

**Explanation:**

Maximum amount he can get by looting 1st, 3rd and 5th house.

Which is  $6+5+4=15$ .

Approach : either take `arr[i]` ya phir `arr[i+1]` aur phir respective call

Imp : dp ka size  $n+3$  tak lio becoz hum `dp[n+3]` k check karenge for  $-1$

Hashing :

1..... Positive Negative Pair: Given an array of distinct integers, find all the pairs having both negative and positive values of a number in the array.

**Input:**  
n = 8  
arr[] = {1,3,6,-2,-1,-3,2,7}  
**Output:** -1 1 -3 3 -2 2  
**Explanation:** 1, 3 and 2 are present pairwise positive and negative. 6 and 7 have no pair.

**Input:**  
n = 3  
arr[] = {3,2,1}  
**Output:** 0  
**Explanation:** No such pair exists so the output is 0.

Approach: Ek set me insert kar then uska negative find kar in set agar hai to result vector me  $\text{abs}(\text{arr}[i])$  and  $-\text{abs}(\text{arr}[i])$  ko insert kr de.  $\text{arr}[i]=0$  ko alag se handle kar liyo

(Sequence maintain karna is the main problem like 1 ka negative pehle mila to usko result me daal de then next one)

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Hashing/problem/positive-negative-pair5209>

2.. SubArray with 0 sum : Given an array of positive and negative numbers. Find if there is a **subarray** (of size at-least one) with **0 sum**.

**Input:**  
5  
4 2 -3 1 6  
**Output:**  
Yes  
**Explanation:**  
2, -3, 1 is the subarray with sum 0.

**Input:**  
5  
4 2 0 1 6  
**Output:**  
Yes  
**Explanation:**  
0 is one of the element in the array so there exist a subarray with sum 0.

2

10 -10

Op- YES

Approach : Prefix Sum karke sabhi elements ki frequency nikal **Since array me 0 nahi hai** if koi bhi element ki frequency 2 hogi to unke beech me subarray sum 0 hoga

AND agr kahi prefix sum 0 aaya tab bhi starting se subarray ka sum 0 hoga

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Hashing/problem/subarray-with-0-sum-1587115621>

3..... Count No. of Subarrays with sum 0:



	0	1	2	3	4	5	6	7	8
Value	-1	2	8	-3	-5	2	-4	6	1
Count	0	2	1	1	2	1	2	1	2

✓ 1 + 1 + 1 + 1 + 2

Approach : [https://www.youtube.com/watch?v=C9-n\\_H7dsvU](https://www.youtube.com/watch?v=C9-n_H7dsvU)

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Hashing/problem/zero-sum-subarrays1825>

4..... Sort an array according to other: Given two integer arrays A1[ ] and A2[ ] of size N and M respectively. Sort the first array A1[ ] such that all the relative positions of the elements in the first array are the same as the elements in the second array A2[ ].

See example for better understanding.

Note: If elements are repeated in the second array, consider their first occurrence only.

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Hashing/problem/relative-sorting4323>

Approach : Pehle map me frequency store kareng (Sorted chahiye is liye map use karo unordered map nhi) then dusree array se map search kar and acc. To freq result vector me daal then jo Bache hai usse bhi result me add kar

5..... Subarrays with equal 1s and 0s : Given an array containing 0s and 1s. Find the number of subarrays having equal number of 0s and 1s.

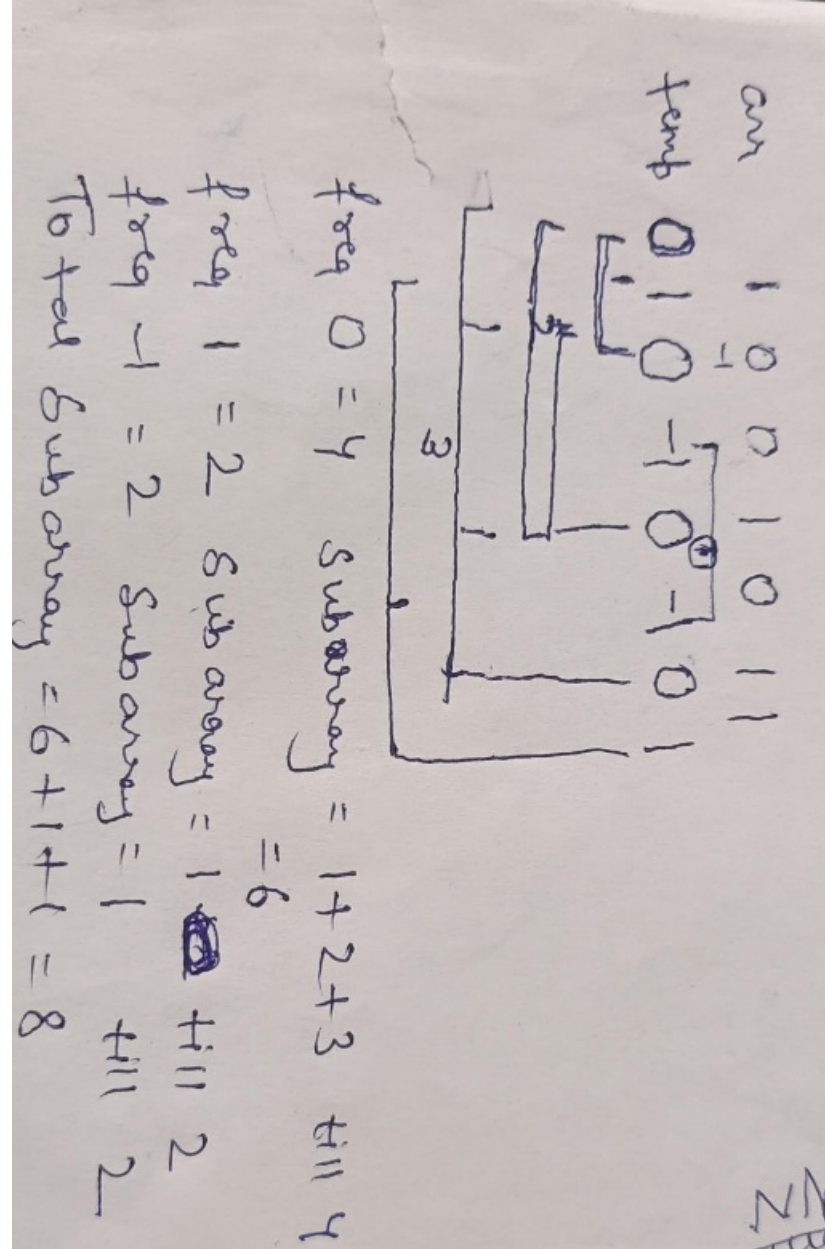
**Input:**  
n = 7  
A[] = {1,0,0,1,0,1,1}

**Output:** 8

**Explanation:** The index range for the 8 sub-arrays are: (0, 1), (2, 3), (0, 3), (3, 4), (4, 5), (2, 5), (0, 5), (1, 6)

Approach : Since 1 and 0 hi hai to bas 0 to -1 assume kar de and initial me prefix sum 0 liyo then jaha bhi vo repeat hoga uss sub array ka 0 and count equal hoga. Pehla element 0 lene se first se end takagr sum 0 hua to uss case ko handle karega.





<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Hashing/problem/count-subarrays-with-equal-number-of-1s-and-0s-1587115620>

6.... Subarray range with given sum : Given an unsorted array of integers and a **sum**. The task is to count the number of subarray which adds to the given sum.

#### Input:

n = 5

arr[] = {10,2,-2,-20,10}

sum = -10

**Output:** 3

**Explanation:** Subarrays with sum -10 are:

[10, 2, -2, -20], [2, -2, -20, 10] and  
 [-20, 10].

#### Input:

n = 6

arr[] = {1,4,20,3,10,5}

sum = 33

**Output:** 1

**Explanation:** Subarray with sum 33 is:  
 [20,3,10].

Approach : Create a prefix sum array and then search for frequency of prefix sum-k(which sum to find) . and go according to 5<sup>th</sup> problem.

<https://practice.geeksforgeeks.org/batch/dsa-4/track/DSASP-Hashing/problem/subarray-range-with-given-sum0128>