

PREREQUISITE (course)

## TOPIC-1: HASHING

1.Name:Query Count Number

Code link: <https://www.jdoodle.com/ia/1G0O>

Desc :read input then use  $O(n)$  loop to find out answer for each query

2.Name:Frequency Of element in array

Code link: <https://www.jdoodle.com/ia/1G0P>

Desc :Used hashmap to hash all values and then return ans for target

3.Name:Max frequent and min freq element in array

Code link: <https://www.jdoodle.com/ia/1G0Q>

Desc :Used hashmap to hash all values and then return use two variables to track max min vars .

4.Name:check if two elements are same in k range

Code link: <https://www.jdoodle.com/ia/1G0R>

Desc :Used hashmap to store indices and checked everytime where the current element existed and if yes then the difference in indices  $\Rightarrow$  yes ? true : continue to search

5.Name: count pairs ( $b[j] + b[i] == k$ )

Code link: <https://www.jdoodle.com/ia/1G0V>

Desc :Used hashmap to store indices .  $i + j = \text{target}$  . if at every point we have i element then its complimentary can be found in hassmap using  $m[\text{target}-j]$  and count if present

6.Name: count pairs ( $b[i] - b[j] == k$ ) ( same question but sign change)

Code link: <https://www.jdoodle.com/ia/1G0Y>

Desc :Used hashmap to store indices .  $i - j = \text{target}$  . if at every point we have i element then its complimentary can be found in hassmap using  $m[\text{target}+j]$  and count if present . twist - we check from end to ensure  $\Rightarrow i < j$  (search right side )

7.Name: count pairs  $i < j$  and  $\text{abs}(b[i] - b[j]) = k$  [ $k \geq 0$ ]

Code link: <https://www.jdoodle.com/ia/1G0Z>

Desc :Used hashmap to store indices .  $i - j = \text{target}$  . if at every point we have i element then its complimentary can be found in hassmap using  $m[\text{target}+j]$  and  $m[\text{target}-i]$  since it is absolute value and add count.

9.Name: find sum of range[l to r inclusive]

Code link: <https://www.jdoodle.com/ia/1G12>

Desc :prefix sum to maintain 1 based index sum array and then for l to r range add r range elements - l-1 range elements

10.Name: find no of subarray with sum ==k

Code link: <https://www.jdoodle.com/ia/1G13>

Desc :prefix sum to maintain 1 based index sum array and then for l to r range add r range elements - l-1 range elements

11.Name: Find largest/smallest subarray with sum k in Given Array

Code link: <https://www.jdoodle.com/ia/1G14>

Desc : use hashmap to find to store indices to find subarray and then use vars to track min max length

12.Find count of shortest/largest subarrays with sum k in given array

Code link: <https://www.jdoodle.com/ia/1G17>

Desc : use hashmap to find to store indices to find subarray and then use vars to track min max length and else condition to update their counts

13. Valid Anagrarm

Code link: <https://www.jdoodle.com/ia/1G19>

Desc : use hashmap to find to store indices to find subarray and then use vars to track min max length and else condition to update their counts

**14. Given an array find two subarrays of maximum sum which are not intersecting**

Code link: <https://www.jdoodle.com/ia/1G1E>

Desc : used kadane algorithm + partition to find out max subarrays sum

**15. Two sum**

Code link: <https://www.jdoodle.com/ia/1G1E>

Desc : Use map to find complimentary and then return as soon as it is found

**16. Cumulative sum query**

Code link: <https://www.jdoodle.com/ia/1G1I>

Desc : using standard prefix sum to track query range sums

**17. Max Distance Between Two Occurrences**

Code link: <https://www.jdoodle.com/ia/1G1J>

Desc : using hashmap to find the previous occurrence and using maxi variable to track maximum diff obtained

**18. First non repeat character**

Code link: <https://www.jdoodle.com/ia/1G1L>

Desc : using frequency array to track whose freq is 1 and return it

**19. Find Common Characters**

Code link: <https://www.jdoodle.com/ia/1G1P>

Desc : Using freq array for all strings and using minFreq to track min freq among each word and appending it to answer

**20. Longest Subarray with sum 0**

Code link: <https://www.jdoodle.com/ia/1G1S>

Desc : hasmap to store the indice and calculate complimentary(for target)

**21. Longest Consecutive Sequence**

Code link: <https://www.jdoodle.com/ia/1G1T>

Desc : sorted array and used maxi to find longest consecutive sequence

**22. Count Number of Pairs With Absolute Difference K**

Code link: <https://www.jdoodle.com/ia/1G1V>

Desc : used hashmap to store the freq and for each i find its compilmentary(for target) exists or not and add count

23. Count Subarrays with sum k

Code link: <https://www.jdoodle.com/ia/1G1W>

Desc: If sum + complimentary == target then increase count

24. Count Subarrays with given XOR

Code link: <https://www.jdoodle.com/ia/1G1X>

Desc: If sum ^ complimentary == target then increase count

25. Zscaler OA

Code link: <https://www.jdoodle.com/ia/1G4c>

Desc: Used pq to convert largest to second largest

TOPIC : Two Pointer

26. Number of subarrays whose sum  $\leq K$

Code link: <https://www.jdoodle.com/ia/1G21>

Desc: use sliding window to count all subarrays

27. Number of subarrays whose distinct element count  $\leq K$

Code link: <https://www.jdoodle.com/ia/1G28>

Desc: use sliding window to count all subarrays + hashmap to track distinct

28. Number of pairs whose diff  $\leq K$

Code link: <https://www.jdoodle.com/ia/1G29>

Desc: sort array and counted the pairs having diff  $\leq k$  using 2pointer

29. Longest Continuous Subarray With Absolute Diff Less Than or Equal to Limit

Code link: <https://www.jdoodle.com/ia/1G2a>

Desc: using sliding window and finding out max and min for every subarray and adding to count

### 30. CF question

(Given an array of  $n$  integers  $a[i]$ . Let's say that a segment of this array  $a[l..r]$  ( $1 \leq l \leq r \leq n$ ) is good if the difference between the maximum and minimum elements on this segment is at most  $k$ . Your task is to find the number of different good segments.)

Code link: <https://www.jddoodle.com/ia/1G2c>

Desc: counted the valid subarrays using sliding window using orderedMap

### 31. Given an array of all integers $\geq 0$ -> find the largest subarray whose sum is $\leq k$

Code link: <https://www.jddoodle.com/ia/1G2d>

Desc: use sliding window to count find the valid subarray and variable to track maxi

### 32. Find the largest valid substring - Valid string is a string where any pair of characters have $\text{diff} \leq k$

Code link: <https://www.jddoodle.com/ia/1G2k>

Desc: used ordered map to find out the largest and smallest element in map using sliding window technique

## TOPIC : Binary Search

### 33. Find First and Last Position of Element in Sorted Array

Code link: <https://www.jddoodle.com/ia/1G2u>

Desc: efficiently used BS to find first occurrence and last occurrence.  
Find element and search in appropriate directions

### 34. Find upperbound for a number in array

Code link: <https://www.jddoodle.com/ia/1G2v>

Desc: used inbuilt iterator

### 35. Search in Rotated Sorted Array

Code link: <https://www.jddoodle.com/ia/1G2z>

Desc: Bs and compared low mid and high to find the pivot and search in appropriate range

### 36. Single Element in a Sorted Array

Code link: <https://www.jdoodle.com/ia/1G2B>

Desc: BS on indexes to find the even odd pair pattern and check where this pattern breaks to search in that range

### 37. N-th root of a number

Code link: <https://www.jdoodle.com/ia/1G2x>

Desc: Used binary search to calculate answer in sorted range of 1 to  $\sqrt[n]{n}$

TOPIC:Greedy

38. Given an array of size "N" ;  $b[i] \rightarrow$  represents the happiness you will get by selecting the  $i$ th element. Task is to select K elements ; and maximize your happiness.

Code link: <https://www.jdoodle.com/ia/1G2D>

Desc: Sort and the answer is last k elements (observations)

39. Given an array of size "N" ;  $b[i] \rightarrow$  represents the happiness you will get by selecting the  $i$ th element. Task is to select K elements ; and minimize your happiness.

Code link: same as above but first k elements

Desc: Sort and the answer is first k elements (observations)

40. Given two arrays a and b  $\rightarrow$  can contain negative elements. Minimize summation of  $a[i]*b[i]$  overall  $i \rightarrow$  you are allowed to re-arrange a and b in any order

Code link: <https://www.jdoodle.com/ia/1G2G>

Desc: intuition + exchange lemma principle for proof

41. Given two arrays a and b  $\rightarrow$  can contain negative elements. Maximize summation of  $a[i]*b[i]$  overall  $i \rightarrow$  you are allowed to re-arrange a and b in any order

Code link: just sort the both array in asc order and return multiplication of all el

Desc: intuition + exchange lemma principle for proof (same as above with slight modification)

### 42. Minimum Amount of Damage Dealt to Bob

Code link: <https://www.jdoodle.com/ia/1G2H>

Desc: Killing the max health enemy dealing max damage is the key

43. IBM OA (Given an array of size N; and a target -> find the minimum number of operations needed to make all elements of array equal to target -> there are multiple target in order of Q)

Code link: <https://www.jdoodle.com/ia/1G4u>

<https://www.jdoodle.com/ia/1G4y>

Desc: Brute force ( $n \cdot q$ ) + optimized using formula+prefix+BS

44. Codechef 151 (shooting version easy)

Code link: <https://www.jdoodle.com/ia/1G4M>

Desc: used above question concept

TOPIC : Recursion

44. Find Sum of all the Numbers from 0 til N.

Code link: <https://www.jdoodle.com/ia/1G2T>

Desc:  $\text{sum}(n) = \text{sum}(n-1) + n$  ; if( $n==0$ ) return 0;

TOPIC : Dynamic Programming

45. We are given an array of integers( $a[n]$ ) . We are given multiple queries of the form : (1, i) which means we need to output the sum of all numbers from index- '1' to index 'i' of the array.

Code link: <https://www.jdoodle.com/ia/1G2U>

Desc: used dp array to find the psum

46. Given an array of integers(positive as well as negative) ,select some elements from this array(select a subset) such that:-

1. Sum of those elements is maximum(Sum of the subset is maximum) .

2. No two elements in the subset should be consecutive.

Code link: <https://www.jdoodle.com/ia/1G36>

Desc: // two choice to make

//either rob the current house or and go to next next house

//or rob the next house and exclude the current

//  $\text{rec}[i] = (\text{cost}[i] + \text{rec}[i+2], \text{rec}[i+1])$

47. Modified Version : We are given '2' arrays . Select some elements from both of these arrays (select a subset) such that:-

--->1. Sum of those elements is maximum(Sum of the subset is maximum).

--->2. No 2 elements in the subset should be consecutive.(Note:-If you select, say the 5th element from Array-1, then you are not allowed to select 4th element from either Array-1 or Array-2 nor are you allowed to select the 5th element from Array -2 all of them are considered consecutive :-) )

1. Sum of those elements is maximum(Sum of the subset is maximum).

2. No two elements in the subset should be consecutive

Code link: <https://www.jddoodle.com/ia/1G32>

Desc:  $dp[i] = \max(\max(a1[i], a2[i]) + dp[i + 2], dp[i + 1]);$

48.A-Frog-1

Code link: <https://www.jddoodle.com/ia/1G39>

Desc:  $dp[i] = \min(\begin{aligned} &dp[i - 1] + \text{abs}(h[i] - h[i - 1]), \\ &dp[i - 2] + \text{abs}(h[i] - h[i - 2]) \end{aligned})$

49. B - Frog 2 (modified version with k max limit jumps)

Code link:<https://www.jddoodle.com/ia/1G3c>

Desc:  $\text{int include} = \text{abs}(\text{arr}[i + j] - \text{arr}[i]) + dp[i + j];$   
 $dp[i] = \min(dp[i], \text{include});$

50.OA Paypal

Code link: <https://www.jddoodle.com/ia/1G3i>

Desc:  $\text{if } (\text{abs}(s[i] - s[i - 1]) \leq k) \{$   
 $\quad dp[i] = dp[i - 1] + 1;$   
 $\quad \} \text{ else } \{$   
 $\quad \quad dp[i] = 1;$   
 $\quad \}$



## TOPIC: MATHS

### 51. Divisors Array (Easy Version)

Code link: <https://www.jddoodle.com/ia/1G5X>

Desc: used the formula of no of divisors and spf relation to find out answer

## TOPIC: Trees

52. Q :-> Given a Tree of "N" nodes and N-1 edges ; rooted at node "1" ; print "N" integers ; where the ith integer prints the number of children of the ith node. Once this is done ; print all the leaves of the particular tree.

Code link: <https://www.jddoodle.com/ia/1G3v>

Desc:

53. Q :-> Google Interview Problem.-> Given a Tree of 'N' Nodes and 'N-1' Edges; rooted at Node-1 ; each node is assigned either 1 or 0 ; for each node "i" ; find the number of 1's on the shortest path from node 1 to node "i"

Code link: <https://www.jddoodle.com/ia/1G3y>

Desc:dfs call (storing as adjlist+ storing ht for each node)

54.Q :-> Q : Given a Tree of "N" nodes; find the height of each node and print it ; the tree is rooted at Node-1.

Code link: <https://www.jddoodle.com/ia/1G3B>

Desc:dfs call (storing as adjlist+ storing ht for each node)

55.Q :->Given a Tree of N Nodes, rooted at node 1 find the sum of each subtree "i" in the given tree.

Code link: <https://www.jddoodle.com/ia/1G3D>

Desc:dfs call (storing as adjlist+ storing sum for each node)

56.Q:>Understanding -> Given a Tree of "N" nodes and "N-1" edges; each node has a value; b[i] -> value of the ith node. Tree is rooted at node 1(1→N) (vm ware OA)

-1000000000<=b[i]<=1000000000

-> Find any path in the tree with a maximum sum such that it only goes downwards.

Code link: <https://www.jddoodle.com/ia/1G3H>

Desc:

TOPIC: Graph

57.Q:>Take the graph; for each node "i" how many nodes are directly connected to it

Code link: <https://www.jdoodle.com/ia/1G3J>

Desc: use adjacency list for graph implementation

58.Q:>BFS for graph

Code link:<https://www.jdoodle.com/ia/1G3V>

Desc:Used a queue to perform level order traversal

59.Q:>Shortest Distance from Source

Code link: <https://www.jdoodle.com/ia/1G3W>

Desc:Used standard bfs for undirected unweighted graph

60.Q:>DFS of graph

Code link:<https://www.jdoodle.com/ia/1G40>

Desc:recursive approach to find the depths of nodes