DSA Round(45min) (https://www.geeksforgeeks.org/amazon-interview-set-27/)

https://www.geeksforgeeks.org/problems/remove-duplicates-in-small-prime-array/1

Given an array consisting of only prime numbers, remove all duplicate numbers from it.

Example 1:

Input:

N = 6

A[] = {2,2,3,3,7,5}

Output: 2 3 7 5

Explanation: After removing the duplicate

2 and 3 we get 2 3 7 5.

Your Task:

Complete the function removeDuplicate() that takes the array of integers and N as input parameters and returns the modified array which has no duplicates. Retain only the first occurrence of the duplicate element. The elements in the returning array should be in the same order as they appear in the original array.

Expected Time Complexity: O(N).
Expected Auxiliary Space: O(1).

Constraints:
1<=N=106
2<=A[i]<100

Your Solution:(10/20)

1. didn't ask for constraints.
2. Start writing code without confidence in logic.
3. Not impress with first 2 approach(very bad and random solutions)

```
Int main()
{
Int n = 6;
vector<int> arr = {2,2,3,3,7,5};
map<int, bool> mp;
for(auto i&&: arr)
{
Mp[i] = true;
}
vector<int> ans;
for(int i=0;i<n;i++){
if(mp[arr[i]]==true){
ans.push_back(arr[i]);
mp[arr[i]]=false;
}
}
```

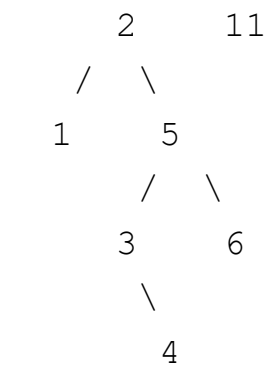
https://www.geeksforgeeks.org/problems/find-the-closest-element-in-bst/1

Given a **BST** and an integer. Find the least absolute difference between any node value of the BST and the given integer.

Example 1:

Input:

```
        10
      /    \
```



K = 13

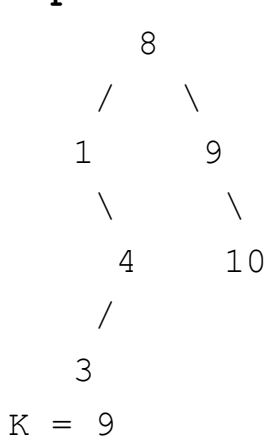
Output:

2

Explanation: K=13. The node that has value nearest to K is 11. so the answer is 2

Example 2:

Input:



K = 9

Output:

0

Explanation: K=9. The node that has value nearest to K is 9. so the answer is 0.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **minDiff()** that takes the root of the BST and an integer K as its input and

returns the minimum absolute difference between any node value of the BST and the integer K.

Expected Time Complexity: O(Height of the BST).
Expected Auxiliary Space: O(Height of the BST).

Constraints:
1 <= Number of nodes <= 10^5
1 <= Value stored at nodes(data), K <= 10^5

Your solution:(10/30)

1. Taking much more time in understading problem and not reading it carefully.
2. Brute force is correct could not optimize.
3. Murmuring in hindi that is not impressive.
4. Lake of confidence with optimize solution.
5.
```
Void solve(Node *root, int k, int &mini)
{
if(!root)
Return;
Mini = min(mini, abs(k-root->data));
solve(root->left, k, mini);
solve(root->right, k, mini);
}

 int minDiff(Node *root, int K)
    {
   if(!root)
   Return k;
   Int mini = INT_MAX;
   solve(root, k, mini);
   Return mini;
    }
```

Oops in C++(15min)

Explain class and object?(Not impressed)(1/5)

A class is a building block of Object Oriented Programs. It is a user-defined data type that contains the data members and member functions that operate on the data

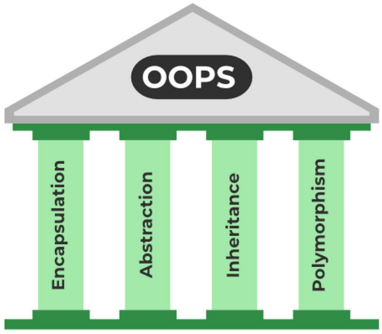
members. It is like a blueprint or template of objects having common properties and methods.

An object is an instance of a class. Data members and methods of a class cannot be used directly. We need to create an object (or instance) of the class to use them. In simple terms, they are the actual world entities that have a state and behavior.

What are the main features of OOPs?(5/5)

The main feature of the OOPs, also known as 4 pillars or basic principles of OOPs are as follows:

1. Encapsulation
2. Data Abstraction
3. Polymorphism
4. Inheritance



What is Inheritance? What is its purpose?(3/5)

The idea of inheritance is simple, a class is derived from another class and uses data and implementation of that other class. The class which is derived is called child or derived or subclass and the class from which the child class is derived is called parent or base or superclass.

The main purpose of Inheritance is to increase code reusability. It is also used to achieve Runtime Polymorphism.

You have to create a E-commerce software with functionalities like User can Add to cart items, remove from cart, how oops will be helpful for designing this software.(0)

Explain multilevel inheritance with a simple example.(4/5)

React coding questions.(45 min)

Implement Redux for a counter variable and create 3 component one for increase the counter , second for decrease the counter and third one for displaying counter variable.(25/40)

We have given an array of intergers you have to calculate $\sum_{i=0}^{i=n-1} (-1)^i \times (array[i])$ **using recursion in javascript.(0/10)**

React Theory(15min)

What is Flux? (no idea 0/5)

What is Redux? (Not concise 3/5)

What are the core principles of Redux? (4/5)

What are the downsides of Redux compared to Flux? (0/5)

What is the difference between mapStateToProps() and mapDispatchToProps()? (0/5)

Can I dispatch an action in reducer? (0/5)