iSBST → left.isBST && right.isBST && myval > left.max && myval < right.mini

maxi → right.maxi
mini → left.mini
sum → left.sum + right.sum + myval
monsum ⇒ Math.Max(sum, left.monsum, right.monsum)

```java
public int minSubarray(int[] A,int n, int p) {
    //Write code here
    int total = 0;
    for(int x:A){
        total = (total+x)%p;
    }
    if(total == 0) return 0; //you dont need to del
    HashMap<Integer,Integer> hm = new HashMap<>();
    hm.put(0,-1); //this is to handle if we are se
    int mini = Integer.MAX_VALUE;
    int pref = 0;

    for(int i = 0;i<n;i++){
        pref = (pref+A[i])%p;
        int key = (pref-total+p)%p;
        if(hm.containsKey(key)){
            mini = Math.min(mini,i-hm.get(key));
        }
        hm.put(pref,i);
    }
    return mini==A.length?-1:mini;
}
```

$P = 9$

$0 \quad 6 \quad 0$

total = 2

$(6 - 7 + 9) \% 9$

$8 \Rightarrow 9$

8

0: -1
6: 0
0: 2