1. class Solution:

    def findCenter(self, edges: List[List[int]]) -> int:

        return list(set(edges[0]) & set(edges[1]))[0]

2. class Solution:

    def minAddToMakeValid(self, s: str) -> int:

        stack=[]

        c=0

        for i in s:

            if i=='(':

                stack.append(i)

                continue

            if i==')':

                if not stack:

                    c+=1

                else:

                    if stack[-1]=='(':

                        stack.pop()

                continue

            c+=1

        return c+len(stack)

3. class Solution:

    def shortestSubarray(self, nums: List[int], k: int) -> int:

        n = len(nums)

        preSum = 0

        queue = collections.deque([(-1,0)])

        res = n + 1

        for i in range(n):

            preSum += nums[i]

            if nums[i] > 0:

                while queue and preSum - queue[0][1] >= k:

                    res = min(res, i - queue.popleft()[0])

            else:

                while queue and preSum < queue[-1][1]:

                    queue.pop()

            queue.append((i, preSum))

        return res if res <= n else -1