Applying these logics may solve your problem

# Think carefully Is this similar to the problem you solved so far?

# Code the optimal solution by checking Time complexity

# Initially reduce your problem size to smaller then finally convert the solution for bigger test cases and edge cases

# Arrays

1. Recursive calls to sub problem
2. Flag variable and break
3. Binary Search
4. Sorting if needed use Key option sorted(arr,key=lambda a: arr[1])
5. Problem involves rotation try **mod** operation.
6. from collections import Counter
7. set() for removing duplicates
8. starting from reverse
9. approaching brute force only but keep tracking with more variables
10. max() ,min() , abs()
11. Dividing the problem by different cases.
12. Separating the cases by even and odd.
13. Keep tracking from left and right side of list
14. Boolean values True and False
15. For multidimensional array keep track the index
16. Hash Table or Dictionary

# DP

1. Sub-solution needed for further computation
2. Memorization technique

# Graphs

1. Bfs , Dfs

# Greedy

1. Ratio sorting fractional Knapsack

# Math’s and Stats

1. GCD , LCM
2. Prime no’s
3. Divide and conquer

# Stacks and Queues

1. FIFO
2. LIFO

# Strings

1. Sub-string
2. isalpha(),upper(),lower()

# Trees

1. Binary Search