# HackerRank Format Challenges

## Next Greater Element

\*\*Description:\*\* Find next greater element for each item in an array. If none, use -1.

\*\*Problem Statement:\*\*  
Given an array arr[] of integers, find the next greater element for each element in order of appearance. The next greater element of an element is the nearest element on the right that is greater. If none exists, use -1.

\*\*Input Format:\*\*  
- A single list of integers arr[]

\*\*Constraints:\*\*  
1 ≤ len(arr) ≤ 10^5, -10^9 ≤ arr[i] ≤ 10^9

\*\*Output Format:\*\*  
- A list of integers representing next greater elements

\*\*Tags:\*\* stack, array, monotonic-stack

\*\*Sample Test Cases:\*\*

1. Test Case 1:

Input: [1, 3, 2, 4]

Output: [3, 4, 4, -1]

1. Test Case 2:

Input: [6, 8, 0, 1, 3]

Output: [8, -1, 1, 3, -1]

1. Test Case 3:

Input: [10, 20, 30, 50]

Output: [20, 30, 50, -1]

1. Test Case 4:

Input: [50, 40, 30, 10]

Output: [-1, -1, -1, -1]

1. Test Case 5:

Input: [5, 1, 7, 2, 6]

Output: [7, 7, -1, 6, -1]

## Time Needed to Buy Tickets

\*\*Description:\*\* Simulate ticket buying queue to find time for k-th person.

\*\*Problem Statement:\*\*  
Given a list tickets where tickets[i] is the number of tickets person i wants, return the time taken for person k to finish buying all their tickets. Each person buys 1 ticket at a time and rejoins the end of queue.

\*\*Input Format:\*\*  
- tickets: List[int]  
- k: Integer index (0-based)

\*\*Constraints:\*\*  
1 ≤ len(tickets) ≤ 100, 1 ≤ tickets[i] ≤ 100, 0 ≤ k < len(tickets)

\*\*Output Format:\*\*  
- An integer representing total time in seconds

\*\*Tags:\*\* queue, simulation

\*\*Sample Test Cases:\*\*

1. Test Case 1:

Input: tickets = [2,3,2], k = 2

Output: 6

1. Test Case 2:

Input: tickets = [5,1,1,1], k = 0

Output: 8

1. Test Case 3:

Input: tickets = [1,1,1,1], k = 2

Output: 3

1. Test Case 4:

Input: tickets = [4,2,1,3], k = 3

Output: 10

1. Test Case 5:

Input: tickets = [3,2,1,4], k = 1

Output: 8

## Minimum Number of K Bit Flips

\*\*Description:\*\* Find the minimum number of k-bit flips to make all bits 1.

\*\*Problem Statement:\*\*  
Given a binary array nums and an integer k, return the minimum number of k-bit flips needed to convert all 0s to 1s. If impossible, return -1. A k-bit flip inverts each bit in a contiguous subarray of length k.

\*\*Input Format:\*\*  
- nums: List[int], binary values  
- k: Integer, flip size

\*\*Constraints:\*\*  
1 ≤ len(nums) ≤ 10^5, 1 ≤ k ≤ len(nums)

\*\*Output Format:\*\*  
- Minimum number of flips or -1 if not possible

\*\*Tags:\*\* greedy, sliding-window, deque

\*\*Sample Test Cases:\*\*

1. Test Case 1:

Input: nums = [0,1,0], k = 1

Output: 2

1. Test Case 2:

Input: nums = [1,1,0], k = 2

Output: -1

1. Test Case 3:

Input: nums = [0,0,0,1,0,1,1,0], k = 3

Output: 3

1. Test Case 4:

Input: nums = [0,0,1,0,1], k = 2

Output: 3

1. Test Case 5:

Input: nums = [1,0,0,1,0], k = 3

Output: 2