Hope you're having some extra fun!

Sub-Problem 1 : Longest Consecutive Sequence

Given an unsorted array of integers, find the length of the longest consecutive elements sequence.

- Simple Test Case :

Input: [100, 4, 200, 1, 3, 2]

Output: 4

- Explanation:

Longest Consecutive Sequence The longest consecutive elements sequence in [100, 4, 200, 1, 3, 2] is [1, 2, 3, 4], which has a length of 4.

- Constraints: The array will have at most 10^5 elements. Each element is an integer that will be at most 10^9.

Sub-Problem 2: Maximum Sum Subarray

Given an integer array nums, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.

- Simple Test Case:

Input: [-2,1,-3,4,-1,2,1,-5,4]

Output: 6

- Explanation:

Maximum Sum Subarray The contiguous subarray with the largest sum in [-2,1,-3,4,-1,2,1,-5,4] is [4,-1,2,1], which has a sum of 6.

- Constraints: The array will have at most 10^5 elements. Each element is an integer that will be at most 10^9.

Sub-Problem 3: Sliding Window Maximum

Given an array nums, there is a sliding window of size k which is moving from the very left of the array to the very right. You can only see the k numbers in the window. Each time the sliding window moves right by one position. Return the max sliding window.

- Simple Test Case:

Input: nums = [1,3,-1,-3,5,3,6,7], k = 3

Output: [3,3,5,5,6,7]

- Explanation:

Sliding Window Maximum For the array nums = [1,3,-1,-3,5,3,6,7] with k = 3, the maximum values in each sliding window of size 3 are as follows:

Window [1, 3, -1] -> Maximum is 3

Window [3, -1, -3] -> Maximum is 3

Window [-1, -3, 5] -> Maximum is 5

Window [-3, 5, 3] -> Maximum is 5

Window [5, 3, 6] -> Maximum is 6

Window [3, 6, 7] -> Maximum is 7

So, the output is [3,3,5,5,6,7].

- Constraints: The array will have at most 10^5 elements. Each element is an integer that will be at most 10^9. The window size k will be at most the size of the array.

Main-Problem: The Treasure Hunt

In a far-off land, there is a treasure hunt game. The game is played on a grid of size **n x n.** Each cell in the grid contains a number. The number represents the amount of treasure in that cell. However, there's a catch. The treasure hunters can only move right or down, starting from the top-left cell and ending at the bottom-right cell. Moreover, they can only move to a cell if the number in that cell is part of a consecutive sequence with the number in the current cell. For example, if a cell contains the number 5, the hunters can only move to a cell that contains either 4 or 6.

The goal of the treasure hunters is to collect as much treasure as possible. However, they also have a magic power. They can use this power to increase the treasure in any one cell by **k**. They can only use this power once.

Your task is to help the treasure hunters. Given the grid and the value of **k**, you need to determine which cell they should use their magic power on, and what path they should take to collect the maximum amount of treasure.

Goal: Implement a solution that helps the treasure hunters collect the maximum amount of treasure.

- Simple Test Case:

Input:

grid = [

[1, 2, 3],

[6, 5, 4],

[7, 8, 9]

],

k = 2

Output:

cell = (1, 0),

path = [(0, 0), (0, 1), (0, 2), (1, 2), (2, 2)],

treasure = 29

- Explanation:

The treasure hunters should use their magic power on the cell at (1, 0) to increase the treasure from 6 to 8. Then, they should take the path that goes right from the top-left cell to the top-right cell, then down to the bottom-right cell. This path allows them to collect a total of 29 treasure.

- Constraints: The grid will have at most **100 x 100 cells**. Each cell will contain a positive integer that will be at most 10^9. The value of **k** will be a positive integer that will be at most 10^9.

Good luck!