Problem 1802: Maximum Value at a Given Index in a Bounded Array

Problem Information

Difficulty: Medium
Acceptance Rate: 38.86%
Paid Only: No
Tags: Math, Binary Search, Greedy

Problem Description

You are given three positive integers: `n`, `index`, and `maxSum`. You want to construct an array `nums` (**0-indexed**)**** that satisfies the following conditions:

* `nums.length == n` * `nums[i]` is a **positive** integer where `0 <= i < n`. * `abs(nums[i] - nums[i+1]) <= 1` where `0 <= i < n-1`. * The sum of all the elements of `nums` does not exceed `maxSum`. * `nums[index]` is **maximized**.

Return `nums[index]`_of the constructed array_.

Note that `abs(x)` equals `x` if `x >= 0`, and `-x` otherwise.

Example 1:

Input: n = 4, index = 2, maxSum = 6 **Output:** 2 **Explanation:** nums = [1,2,_**2**_ ,1] is one array that satisfies all the conditions. There are no arrays that satisfy all the conditions and have nums[2] == 3, so 2 is the maximum nums[2].

Example 2:

Input: n = 6, index = 1, maxSum = 10 **Output:** 3

Constraints:

* `1 <= n <= maxSum <= 109` * `0 <= index < n`

Code Snippets

C++:

```
class Solution {
public:
int maxValue(int n, int index, int maxSum) {

}
};
```

Java:

```
class Solution {
public int maxValue(int n, int index, int maxSum) {

}
}
```

Python3:

```
class Solution:
def maxValue(self, n: int, index: int, maxSum: int) -> int:
```