

Jump Game

For solving the classic "Jump Game" problem, the goal is to determine whether you can reach the last index of the array given jump lengths.

Strategy: Greedy (Best and Fastest):

Track the farthest index you can reach (maxReach).

At each index i , check if i is within reach, and update $\text{maxReach} = \max(\text{maxReach}, i + \text{nums}[i])$.

If at any point $i > \text{maxReach}$, it means you can't even get to i \rightarrow return false.

Algorithm:

1. Initialize $\text{maxReach} \leftarrow 0$
2. Loop i from 0 to $n-1$
 - If $i > \text{maxReach} \rightarrow$ return false
 - Update: $\text{maxReach} = \max(\text{maxReach}, i + \text{nums}[i])$
 - If $\text{maxReach} \geq n-1 \rightarrow$ return true
3. After the loop, return true

Time Complexity: $O(n)$ - single scan
 $O(1)$ - constant space

Example:

1. $[2, 3, 1, 1, 4]$

Reachable indices: $0 \rightarrow 1 \rightarrow 4$ ✓

2. $[3, 2, 1, 0, 4]$

You get stuck at index 3 where $\text{nums}[3] = 0$ ✗