The Algorithm Design Canvas

Problem name: Jumping on the Clouds

HIRED IN TECH

Constraints

* jump only on safe clouds
* Emma can jump 2 or 1 positions to reach a safe cloud
* we receive 2 inputs: n -> length, c -> clouds
* c.length === n
* 2<= n <= 100
* c[i] -> only 0 or 1
* c[0] and c[n -1] always 0 ALWAY POSSIBLE TO WIN!

Ideas

* check inputs validity -> fit constraints * counter = 0 * while loop until counter < n - 1 * verify longer distance (+2) * if is a safe cloud -> move to counter safe position: counter + 2(check length) * if not a safe cloud -> move to counter + 1 * no matter what we increase jumps++	O(log(n)) or O(n) if we verify array elements O(n)

Test Cases

* invalid inputs: 0
* n: 4, c: [0,1,0,0] -> 2
* n: 8, c: [0,1,0,1,0,0,1,0] -> 4
* n: 3 c: [0,1,0]
* really big array of clouds

Code