

Problema della Maggioranza: Chiediamo a ChatGPT!

PROMPT

Describe in pseudocode a linear time algorithm for finding the absolute majority in an array of size n

```
function findAbsoluteMajority(A[1..n]):  
    candidate ← null  
    count ← 0  
  
    // Phase 1: Find potential candidate  
    for i from 1 to n:  
        if count == 0:  
            candidate ← A[i]  
            count ← 1  
        else if A[i] == candidate:  
            count ← count + 1  
        else:  
            count ← count - 1
```

```
// Phase 2: Verify candidate
count ← 0
for i from 1 to n:
    if A[i] == candidate:
        count ← count + 1

if count > n/2:
    return candidate
else:
    return "No absolute majority"
```

Complexity

- **Time:** $O(n)$ (two linear passes)
 - **Space:** $O(1)$
-

Example

For `A = [2, 2, 1, 1, 1, 2, 2]`:

- Candidate found = 2
- Count of 2 = 4 > $7/2$ →  majority = 2

Correttezza

Esempio: $P = 8$

A =	③	1	①	2	④	4	4	6	Candidate Max
									4 No
A =	④	1	④	3	④	4	4	6	
									4 SI

Esercizio: dimostrare che alla fine del primo ciclo for , se esiste un elemento di maggioranza in A esso è candidato. (Usare un opportuno invariante)

PROPRIETÀ 2

INVARIANTE che vale alla fine delle iterazioni $i \geq 0$

$A[1:i]$ contiene

* count occorrenze di candidato $(\text{count} > 0)$

* $(i - \text{count})/2$ copie (x, y) $x \neq y$

N.B. queste coppie possono contenere
candidati, le cui occorrenze
possono quindi essere $> \text{count}$





