PRACTICE CERTIFICATION NEW COMPETE JOBS LEADERBOARD Search hackerrank682

Practice > Algorithms > Implementation > Jumping on the Clouds > Editorial

Jumping on the Clouds ☆

1317.53 more points to get your next star!

Rank: 140642 | Points: 882.47/2200

Problem Submissions Leaderboard Discussions Editorial

Editorial by Shafaet

This is a simulation problem. Because the problem guarantees that it is always possible to win, we know that our input will never contain 2 consecutive thunderclouds. To reach the last cloud in a minimum number of steps, always try make a jump from i to $i + 2$. If that is not possible, jump to $i + 1$.

The Problem Setter's solution below uses this approach. Check out the Problem Tester's solution for a slightly different approach.

Set by Shafaet

Problem Setter's code:

Python 2

```
n = int(raw_input())
c = map(int, raw_input().split(" "))
ans = 0
i = 0
while i < n - 1:
    if i + 2 >= n or c[i + 2] == 1:   # Not possible to make a jump of size 2
        i = i + 1
        ans = ans + 1
    else:
        i = i + 2
        ans = ans + 1
print ans
```

Ruby

```
#!/bin/ruby

n = gets.strip.to_i
c = gets.strip
c = c.split(' ').map(&:to_i)
ans = 0
i = 0
while i < n - 1
    (i + 2 >= n or c[i + 2] == 1 )? i=i+1: i=i+2
    ans = ans + 1
end
print ans
```

Tested by ma5termind

Problem Tester's code:

C++

```
#include <bits/stdc++.h>
using namespace std;

const int inf = 555;
int A[111], dp[111];

int main() {
    int n; cin >> n;

    for(int i=1; i<=n; i++) {
        cin >> A[i];
```

STATISTICS

Difficulty:	Easy
Time Complexity:	
Required Knowledge:	Array, Loops
Publish Date:	Apr 20 2016

Originally featured in World Codesprint April

NEED HELP?

View discussions

View top submissions

```
    }
    for(int i=2; i<=n; i++) {
        if(A[i] == 0) dp[i] = min(dp[i-1], dp[i-2]) + 1;
        else dp[i] = inf;
    }
    cout << dp[n] << "\n";
    return 0;
}
```

Feedback

Was this editorial helpful?

Yes No

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature