My Leetcode

Saturday, January 25, 2014

Jump Game II (java)

LeetCode

Given an array of non-negative integers, you are initially positioned at the first index of the array.

Each element in the array represents your maximum jump length at that position.

Your goal is to reach the last index in the minimum number of jumps.

For example:

Given array A = `[2,3,1,1,4]`

The minimum number of jumps to reach the last index is `2`. (Jump `1` step from index 0 to 1, then `3` steps to the last index.)

Solution: Greedy

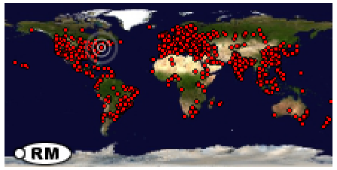
At first, I try to solve this problem with DFS, but exceeded the time limitation, then I search the Internet find a very good solution for this question - Greedy Algorithm. the main idea is try to find the longest distance by each jump can reach and check if this distance can pass the total length of this array, of course we should have a variable to keep record of the current steps. if this distance cannot pass the total length of this array, then we should go through all the position within this distance to see if it can pass the array by jumping from there

I think code should be more clear than my description

Visits Map

```
/*
Given an array of non-negative integers, you are initially positioned at the first index of the array.
```

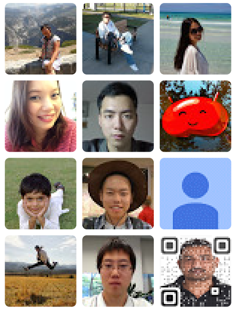
```

Each element in the array represents your maximum jump length at that position.

Your goal is to reach the last index in the minimum number of jumps.

For example:
Given array A = [2,3,1,1,4]

The minimum number of jumps to reach the last index is 2. (Jump 1 step from index 0 to 1, then 3 steps to the last index.)
*/


public class JumpGameII {
public int jump(int[] A) {
if (A==null||A.length==0){
return -1;
        }

if (A.length==1){
return 0;
        }

int minStep=0;

int start=0;
// current longest distance the jump can reach
int end=A[start];

// if current longest distance plus current postion passed the length of array
// then return current minStep + 1;
if (start+end>=A.length-1){
return minStep+1;
        }

while(end<A.length-1){
            minStep++;

// record farest position can be reached by jump from position within current farest position
int max=0;

for (int i=start; i<=end; i++){
int current=i+A[i];
// pass the total length, return minStep+1,

if (current>=A.length-1){
return minStep+1;
```



Google+ Followers

Xinghua Li

Add to circles



84 have me in circles View all

3

Popular Posts

Word Ladder (Java)

Path Sum (Java and Python)
Given a binary tree and a sum, determine if the tree has a root-to-leaf path such that adding up all the values along the path equals the g...

Rotate List (Java)
LeetCode Given a list, rotate the list to the right by k places, where k is non-negative. For example: Given 1->2->3-&...

Labels

BFS Binary Search DFS Divide and Conquer DP Greedy Helper table Leetcode ListNode Recursion Stack Tree Two Pointers

```
                }

                max=Math.max(max, current);
            }
// update start position(items in array are not negative, so end+1 is exist)
            start=end+1;
// update the most far position can reached for next jump
            end=max;
        }

return minStep;
    }
}
```

JumpGameII.java hosted with by **GitHub** **view raw**

Posted by Xinghua Li at 3:31 PM +1 Recommend this on Google

Labels: Greedy, Tw o Pointers

Blog Archive

- ▼ 2014 (102)
  - ► September (1)
  - ► July (1)
  - ► June (12)
  - ► May (2)
  - ► April (2)
  - ► March (6)
  - ► February (29)
  - ▼ January (49)
    - Palindrome Partitioning (Java)
    - Edit Distance (Java)
    - Reverse Nodes in k-Group (Java)
    - Longest Substring Without Repeating Characters (Ja...
    - Distinct Subsequences (Java)

6 comments Google+

Add a comment

Top comments

Xinghua Li shared this via Google+ 2 years ago - Shared publicly

+1 · Reply

Xinghua Li shared this via Google+ 2 years ago - Shared publicly

+1 · Reply

Xinghua Li shared this 2 years ago - LeetCoder (☒☒)

+1

Xinghua Li shared this 2 years ago - Algorithms:Design & Analysis (Discussion)

+1

Xinghua Li shared this 2 years ago - Algorithms & Coding (Puzzles)

+1

Xinghua Li 2 years ago - Shared publicly

Jump Game II (java)

LeetCode Given an array of non-negative integers, you are initially positioned at the first index of the array. Each element in the array represents your maximum jump length at that position. Your goal is to reach the last index in the minimum number of jum...

+1 · Reply

- Combination Sum II (Java)
- Jump Game II (java)
- Add Two Numbers (Java)
- Sqrt(x) (Java)
- First Missing Positive (Java)
- Rotate List (Java)
- Valid Palindrome (Java)
- Scramble String (Java)
- Maximal Rectangle (Java)
- Copy List with Random Pointer (Java)
- Insert Interval (Java)
- Implement strStr(). Java
- Sudoku Solver (Java)
- Clone Graph (Java)
- Merge Intervals (Java)
- Restore IP Addresses (Java)
- Valid Number (Java)
- Largest Rectangle in Histogram (Java)
- Spiral Matrix (Java)
- LeetCode Word Search (Java)
- Best Time to Buy and Sell Stock (Java)
- Multiply Strings Java
- Sort List (Java)
- Binary Tree Maximum Path Sum (Java)
- Simplify Path (Java)
- Minimum Window Substring (Java)
- Substring with Concatenation of All Words (Java)
- Gas Station (Java)
- Candy (Java)
- Word Ladder (Java)
- Interleaving String (Java)
- Decode Ways (Java)
- Max Points on a Line(Java)
- 4 Sum (Java)
- Rotate Image (Java)
- Surround Regions (Java)
- Word Break
- public class Solution { public boolean isInterl...
- String to Integer (atoi)

Newer Post

Home

Older Post

Subscribe to: Post Comments (Atom)

Longest Valid Parentheses (Java)

Divide Two Integers

Single Number II

Single Number

Sort List

► 2013 (13)