Project 2: Goodland Electricity
Submission time: around 11:25 pm. 4/19/2017
Account: shengwei_peng

Algorithm Idea: start from a city $\mathbf{C}_i$ (equal to current + dis where current start from 0)which will cover the neighboring city if it has a tower. If the city does not have a tower then check the previous one, which is $\mathbf{C}_{i-1}$, loop until find a tower and start from that. Then, check the next city, the length should be current city plus cover range plus k. In addition to it, we should have conditions that treat for special case like if the next pointing city is out of the list and others. If no tower found, then return -1.

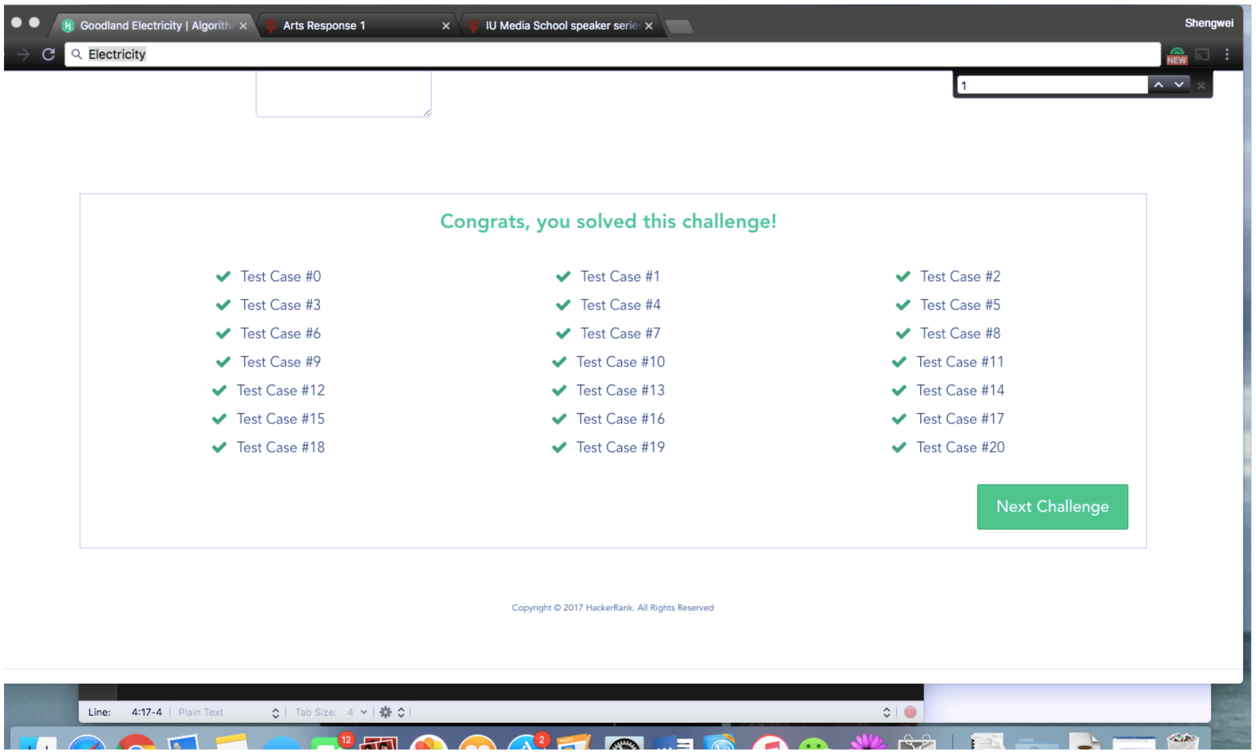
Algorithm explanation:
variable:
n: stands for city
k: stands for range
a[]: list to store city element
dis: cover range
count: count the tower switch on
current: current city

```
First scan the value for n and k
Then scan and copy the rest of elements into list a          →    (time and space : O(n) and n)
Set dis = k – 1, Since the cover range include the city itself

Set count to 0

current starts from 0
While current < n
   set temp = dis
   If the next city, which need to be checked, is over the list range,     -> relocate if out of list
                      then set position to the last city  of the list
   while next city, which need to be checked, has no tower and the pointing city no out of list
        go back and check the previous one city          -> if no city, find one in previous
   if "all city has no tower" or "it goes back and point to the last tower which has switch on and
the next city has no tower"            ->  (two failure cases)
       set count = -1 and break
  else
       tower count should add 1
       set current to next pointing city where current += temp + k
close while loop
print count value
```

for the part above, the worst case will be O(n) and since it accesses the list and do the calculation, it doesn't take much space



Congrats, you solved this challenge!

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2
✓ Test Case #3 ✓ Test Case #4 ✓ Test Case #5
✓ Test Case #6 ✓ Test Case #7 ✓ Test Case #8
✓ Test Case #9 ✓ Test Case #10 ✓ Test Case #11
✓ Test Case #12 ✓ Test Case #13 ✓ Test Case #14
✓ Test Case #15 ✓ Test Case #16 ✓ Test Case #17
✓ Test Case #18 ✓ Test Case #19 ✓ Test Case #20

Goodland Electricity | Algorith
Arts Response 1
IU Media School speaker serie
Shengwei
Electricity
Next Challenge
Copyright © 2017 HackerRank. All Rights Reserved
Line: 4:17-4
Plain Text
Tab Size: 4