

Assignment 2A

Author: Isaac Archer

Due Date: July 14th 13:00 2015

Problem Description: \exists a set $a \in \{1, 0\} \mid |a| = n \in \mathbb{Z} \mid \exists l \in \mathbb{Z} \mid l < n \mid \exists$ a set $b \in \{1\} \mid |b| = l$

Algorithm: First Long 1-String

input: A one based array arr of n elements, with the elements drawn from the set $\{0, 1\}$. An integer lower bound lb .

output: the index of the first position in arr such that there is a string of lb 1s starting at that position. That is: a position i in arr such that $a[i..i+m]$ are all 1s and there is no position $j < i$ in arr with this property. if there is no string of lb 1s in arr , fail.

```
def first-long  $\leftarrow arr, lb$  //  $O(n)$ 

     $cur\_i \leftarrow -1$  // value to show failure to find first long 1-string

    for  $i \leftarrow 1..|arr|$  //  $O(n)$ 
        if  $arr[i] = 1$ 
             $cur\_i \leftarrow i$ 
             $k \leftarrow i$ 
            for  $j \leftarrow k..(k + lb - 1)$  //  $O(n)$ 
                 $i += 1$ 
                if  $arr[j] = 1$ 
                    continue
                else
                     $cur\_i \leftarrow -1$ 
                    break

    if  $cur\_i = -1$  then  $\uparrow$  fail

     $\uparrow cur\_i$ 
```

Correctness

- Returning the Correct Value

The first array starts the search until it hits a one and then stores that current i value. Then it scrolls along from that point and increments the outer loop counter appropriately until it has either counted lb number of 1s or it breaks to the outer loop and sets current i to a failure value. this prevents returning an inaccurate value or keeping the main loop counter from lagging behind and allowing an incorrect count for a false return value.

- Halting

the algorithm will return a fail if the current i values is set to a fail value of -1 which is an invalid index for an array. if it doesn't then there is a default return for the current i value which gives the correct index back.

Worst Case

The worst case is $O(n)$ since the inner loop increments the counter for the outer no more than n comparisons are made. so assuming the basic operation is the compare and or assignment the worst case is $O(n)$.

Best Case

The best case would occur if the lb number of 1s were seen sequentially at the beginning of arr. this would give a best case of $\Omega(lb)$.