

Davie Truong  
I have read and agree to the collaboration policy. Davie Truong  
Homework Heavy

## CMPS 102 — Spring 2017 – Homework 2

### Solution to Problem 2

#### Algorithm

---

```
upper = n
lower = 1
playGame(upper, lower)
    x = floor of median value of (upper, lower)
    y = guess(x) // used to calibrate the median value
    if ( y is correct)
        exit
    else if (guess (x+1) is colder)
        playGame(x, lower)
    else // guess(x+1) is warmer
        playGame(upper, x+1)
```

---

Description: This algorithm works by using the upper, lower, and median value and updating them with each recursive step. At the start of each recursion, the algorithm will get the median value of the upper and lower ranges using floor if there is a float conflict. Then it will call guess on that median value to calibrate the position for when we compare it with the next value. The warm/cold response doesn't matter when we compare with the median value, unless it is a match. So after the median value is calibrated, we guess median + 1 to see how the median's position will be updates for the upper/lower range. If median + 1 is cold that means the median value was closer to the mystery number, making it the new upper limit for the next recursive step. If the median + 1 is warm that means it is closer to the mystery number than the median making it the new lower bound.

*Proof.* Proof of Correctness:

The algorithm will always go for the median value this cutting the search of the list by half every recursive call. This can be understood by observing that the median value replaces the upper or lower limit as the parameters for the recursive call. The algorithm will always be able to reach the highest number or lowest number. Suppose we are left with three values 10(mystery number and upper limit), 9, and 8(lower limit). The algorithm would guess the median number 9 and since 9+1 is warmer, the next recursive step would have (10,10) as the upper/lower parameter this making the median 10, the correct guess, and a way for the algorithm to exit. Suppose we are left with the three values 3(upper limit), 2, and 1(mystery number and lower limit). The algorithm would guess 2 as the median and since 2+1 is colder, the next recursive step would have (2,1) as the upper/lower parameter. In that recursive step, since floor is used for the median in float cases, the median value would be 1, the correct guess, and a way for the algorithm to exit. Since the algorithm checks the median value before updating the upper/lower limit, that means the algorithm is always able to effectively find the number.  $\square$

## Analysis

Time Complexity:

Since the algorithm searches through a list of 1 to  $n$ , and always starts the recursive call at the middle of the list and further splits the list with each call to itself by updating the parameters, the time complexity would be  $\log_2 n$ . With the guesses taking  $O(1)$  time, the final time complexity would be  $\log_2 n + O(1)$

Space Complexity:

Since we initially start with an array of size  $n$  and only work off it, the space complexity would be  $O(n)$