I have read and agree to the collaboration policy. Lynne Diep.

Lynne Diep
Homework Heavy Grading
Collaborators: Sabrina Tsui

Homework 2-2
Due: May 1, 2017

Problem 2:
"Hot and Cold" game - One of you thinks of a number between 1 and n, and the other tries to guess the number. If you are the one guessing, every time you make a guess your sister tells you if you are "warmer", which is closer to the number in her head, or "colder", which is further away from the number in her head.

Using this information you have to come up with an algorithm which helps you guess the number quickly. You can use a command called Guess(x), where x is your guess, which returns "warmer", "colder" or "you guessed it!".

You are required to give an English explanation for your algorithm. Also, prove the correctness of your algorithm and give an analysis of the space and time complexity. An ideal solution will propose an algorithm that takes log2 (n) + O(1) guesses in the worst case scenario.

Solution:
Algorithm -
```
HotandCold(A, low, high)
        i = ⌈(low+high)/2⌉ // initial guess

        response = Guess(i)
        if (response == guess_right)
                end program
                return "You guess it!"
        if (last == i)
                return ⌊(low+high)/2⌋
        last = i
        if (response == warmer)
                if (i > L)
                        HotandCold(A,i,n)
                if (i < L)
                        HotandCold(A,L,n)
        else if (response == colder)
                if (i > L)
                        HotandCold(A,L,n)
                if (i < L)
```

HotandCold(A,i,n)
else(nothing happens)
HotandCold(A,i, high)

English Explanation –
For the array of numbers that can be chosen (1 to n), you find the median of the array, which is $\left\lceil \frac{low+high}{2} \right\rceil$ where low is the first number of the array, and high the last number of the array. With the median, you find out with the response if your number is colder or warmer to the correct number. Which ever the response is, you will find the median of the specific subarray. With that median, you find out if the new number is colder or warmer. This process continues until the correct number is guessed.

Proof of Correctness (by induction) -
Base Case: When k = 1, there is only one element in the array. So that element is the median in the array and the guess is correct.
low = 1 and high = 1 so, $i = \left\lceil \frac{low+high}{2} \right\rceil = \left\lceil \frac{1+1}{2} \right\rceil = \left\lceil \frac{2}{2} \right\rceil = 1$
Induction Hypothesis: Assume p(k) is true for k > 1.
We show that k+1 is true. For array A[1…k+1] we calculate the first guess to be $\left\lceil \frac{1+(k+1)}{2} \right\rceil$, after our initial guess. We will either receive a "guessed correctly" or no response. If the response is "guessed correctly" then the program ends, and we win. If we get no response then we call the algorithm recursively on the subarray. Since the size of the subarray is smaller than A[1…k+1], we use the inductive hypothesis. Therefore, we can find the correct guess. Hence the correctness of our algorithm.

Analysis of Time –
Ideal runtime = $\log_2 n + O(1)$
$T(n/2) + O(1)$
Let a=1 and b=2 $\quad f(n) = O(1) = n^0 = 1$
$n^{\wedge}(\log_2 1) = n^0 = f(n)$
$\theta(n^0 \log n) = \theta(\log n) = \log_2 n + O(1)$

Space Complexity –
The space complexity is n since we only use one array of size 1 to n in the algorithm.