EECS 336 Problem 4.1

Algorithm 1 Peak Interest

```
Require: complete binary T rooted at r
1: function findPeak(T, r)
2:     Set v = r
3:     while True do
4:         if (v.value > v.left.value and v.value > v.right.value) or (v.left == NULL and v.right == NULL) then
5:             return v.value
6:         end if
7:         if v.left.value > v.right.value then
8:             v = v.left
9:         else
10:            v = v.right
11:        end if
12:    end while
13: end function
```

RunTime

The complete binary tree with n nodes has log(n) levels. In the algorithm, there is only one vertex compared on each level. So at most log(n) nodes are compared. Each comparison takes constant time. So the $T(n) = O(log(n))$

Correctness

Claim: Let $V = (r, v_1, v_2, ..., v_m), m \leq logn$ be the sequence of nodes checked by the end of the algorithm. To prove the algorithm is correct, we prove that the last element in this sequence has to a peak.

(Proof by contradiction)

Suppose v_m is not a peak node. Since v_m is the last node in the sequence, then v_m has to satisfy either (1) $v_m.value > v_m.left.value$ and $v_m.value > v_m.right, value$ or (2) $v.left = NULL$ and $v.right = NULL$. For both (1) and (2) According to the definition of peak node, and since v_m is not a peak node, $v_m.value < v_{m-1}.value$ (*).

According to line 7-11 in the algorithm, when v_{m-1} is checked, at least one of its children is greater than it. $\Rightarrow v_{m-1}.value < max\{v_{m-1}.left.value, v_{m-1}.right.value\} = v_m.value$ (according to line 7-11). This contradicts (*). So v_m is a peak node. Therefore, the sequence that the algorithm checked has to contain a peak node.