

## Problem 2

Algorithm:

```
i ← 1
while houses[i] == "keep going":
    i = i * 2
return bsearch( $\frac{i}{2}$ , i)
```

Runtime:

The bounds of  $h$  will always be  $2^k \leq h \leq 2^{k+1}$  where  $k = \lfloor \log_2(h) \rfloor$ .  
The range binary searched will be  $(2^{k+1} - 2^k)$ , so the total runtime would be...  
 $\log_2(2^{k+1} - 2^k) \leq \log(k) + C \rightarrow O(\log(h))$ .

This means the algorithm runs in  $O(\log(h))$  time.

Correctness:

Hypothesis: Returns the index to where houses will return "Party's here!" without making changes to houses.

Base Case:  $h = 1$ , the alg. returns 1. ✓

Induction Step:

Assume:

$\forall i < n$ , it returns  $i$  when  $i = h$ .

The function considers some range  $[i, 2i]$  where  $i = \lfloor \log_2(h) \rfloor$ . We also know this will return the correct result as, we know  $h \geq i$  and  $h \leq 2i$ . Meaning we have not throw out the correct  $h$  and do not have to consider any previous numbers.

This proves that our algorithm will always find the correct  $h$  according to our hypothesis.