Algorithms and Datastructures

Assignment 8

Lucas van der Laan
s1047485

1

(a) $\mathcal{O}(3n)$

(b) $\mathcal{O}(n)$

(c) $\mathcal{O}(n \log n + n)$

(d) $\mathcal{O}(n + n \log n)$

2

We start by splitting all the rooms into two categories, R_1 and R_2. We then make sure that all the switches in R_1 and R_2 are the opposite each other, but the same in the same group. Let's say we start with R_1, we then flip the main switch, if it produces a BANG, we know that the room is in R_1. However, if it does not produce a BANG, we know the room is in R_2. We then repeat this by splitting the culprit group again into R_1 and R_2 and do the same thing, rinse and repeat until we have found our room.
This is $\mathcal{O}(\lg n)$ because we only execute a main switch flip once every level we go down.

3

If n = 1:
We can say that every coordinate that is not the missing coordinate is part of the L-Shape.
Else:
Place an L shaped tile at the center, in the way that it does not cover the missing tile. Divide the square into 4 squares of $2^{\text{currentSize}-1}$ x $2^{\text{currentSize}-1}$, where currentSize starts at n and becomes 1 less every time we do the recursive call. Call above recursively until n = 1, for the base case. The base case returns the placed tile.
We can say this is correct by doing induction based on the base case and the induction hypothesis that if you can solve 1 square you can solve every square of 2^n. The base case says that a square, with a missing hole, of size 2x2 can be solved, so by induction we can say that the hole that we created by placing the L-shaped object in the middle can be solved when we do this of 2^{n-1}.
The running time of the algorithm is:
n = 1: 1
n > 1: $4T(2^{n-1})$, because for every level we have to do 4 operations.

4

I had no idea what the question really asked, I had written down a lot but it was all about the total number of operations, not just the comparisons. With just the comparisons it is pretty clear why it is n^2 because you have a loop with an inner loop comparing against each other.