1. **Determine which wire is connected to each switch, without opening the box**

**Algorithm:**

Main-Function () {

For each wire w from 1 to n

Final-switch = Switch-Connect(w, [s1, … , sn])

Assign Final-switch as switch for wire w

}

Switch-Connect(w, Switch[]) {

If(number of switches in Switch[] is equal to 2 )

Connect wire w and switch on switch[1]

**If bulb glows** Assign Switch[1] as Final-switch

**Else** Assign Switch[2] as Final-switch

Else

Connect wire w and switch on switches s1, … , sn/2

**If bulb glows** Final-switch = Switch-Connect(w, [s1, … , sn/2])

**Else** Final-switch = Switch-Connect(w, [sn/2 + 1 , … , sn])

Return Final-switch

}

**Proof of correction:**

It is mentioned that every wire is connected to exactly one switch

* If bulb doesn’t glow when wire w is connected and switches (s1, … , sn/2) are switched on then the switch for wire w is among (sn/2 + 1 , … , sn)
* For example, if there are only two switches if wire w is connected, if s1 is switched on and bulb doesn’t glow then it is fair to assume that bulb will glow if s2 is switched on

**Proof of Termination:**

Algorithm runs for at most n iterations for n wires and for each wire recursively checks for n switches. No already assigned wire is repeated so the algorithm terminates

**Asymptotic Notation:**

Algorithm will run for n iterations for n wires and each wire w the algorithm to recursively search for the correct switch will take log n times. This overall complexity is O(n log n) and space complexity is O(n)