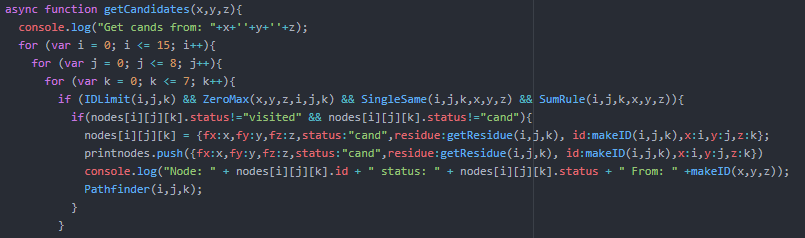
John Scales

Class Number 335-04

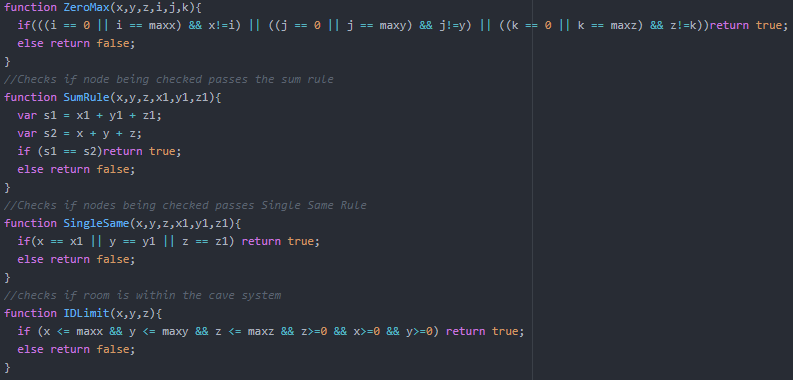
CWID: 888865730

Aswilanaga - Big-O Complexity

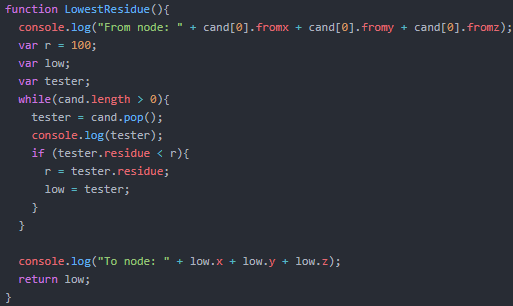
The Big-O Running time of my code is O(N^3). The code gets very complicated and there are a lot of nested for loops. The deepest nesting is 3 for loops which makes my algorithm’s case O(N^3). I am wondering, however, if since N is 15\*8\*7, does that make my algorithm O(N) instead since it is only going through that many nodes (max) at a time?



Above is the main part of the algorithm that checks if a node obeys the conditions given with the project. Inside of the functions there are only comparisons and assignments:

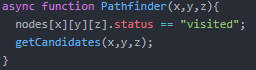


This means that the algorithm remains O(N^3) (or O(N) if that’s what it actually is).

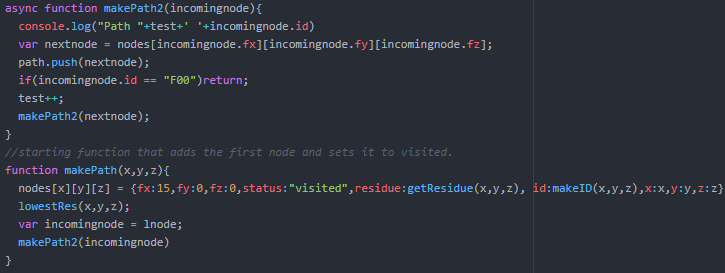


This function (above) finds the lowest residue node in the available nodes and is O(N)

The below is a very basic O(1) function not including the getCandidates complexity.



The rest of my functions are basic and are O(N) or less because they only go through a subset of N (could be all of N, but not in this case so it remains O(N))



Overall after looking at my code and examining the complexity, I would say it’s O(N) since the 3 nested for loops at the beginnning of this analysis only go through N nodes not N^3 nodes.