

CPSC-02, Project #2: Asgwillanga Cavern

Overall Complexity: Breadth-First Search()

Given N for nodes and E for edges. The overall time complexity of the breadth-first search runs at $O(V+E)$ time.

There are several helper functions, such as confirming if each room can be traversed (sum, zero-max, limit, and single-same). Our helper functions, run at $O(1)$ time. We traverse all the nodes to find the smallest residue. Therefore, our search runs at least $O(N)$ time. Furthermore, we must search the “neighbors” of each node, which results in at least $O(E)$ time. We are traversing an undirected graph, and that results in our edges appearing twice in the adjacency list. Overall, the time complexity results in multiple $O(1) + O(N) + O(2E)$. This can be simplified to $(N+E)$.

Helper Functions: generate_nodes() and verify_room()

Once we create our Graph data structure, we must initialize the nodes and edges. Each node and edge is hard-coded in. and some vertices have multiple neighbors. Overall, this is simplified to $O(1)$.

Additionally, we must ensure each cave passes our traversal rules. In order to do so, a helper function `verify_room()` checks if the room can be traversed. Each function runs at $O(1)$; therefore, `verify_all()` runs approximately at $O(1)$ time.