**Lab: Helping Terry the Rat Escape the Underground Pipes**

**1. Problem Statement**

**In this lab, Terry the rat is trapped inside a large underground pipe system. The pipes connect different junctions, and each pipe has a cost, representing the time or difficulty Terry takes to crawl through. Our goal is to help Terry move from his starting junction to the target junction where the cheese is hidden.**

**2. Understanding the Problem**

* **Each junction can be thought of as a node.**
* **Each pipe acts as a connection (edge) with a cost.**
* **Terry’s goal is to find a path from the start junction to the cheese junction.**
* **The start junction is the initial node, and the target junction is the cheese location.**

**Essentially, this is a graph problem, where we want to find paths efficiently.**

**3. Plan and Approach**

**We model the pipe network as a graph with nodes (junctions) and edges (pipes). Then, we use search algorithms to find paths.**

**Inputs:**

* **Number of junctions and pipes**
* **Connections between junctions with travel costs**
* **Coordinates of junctions (for distance calculation)**
* **Start junction and target junction**

**Outputs:**

* **Path Terry takes for each search strategy**
* **Total cost of the path**
* **Number of junctions visited**

**4. Algorithms Used**

1. **Depth First Search (DFS)**
   * **Terry goes deep along one path before backtracking.**
   * **May not find the cheapest path but is simple and fast to implement.**
2. **Breadth First Search (BFS)**
   * **Terry checks all nearby junctions first.**
   * **Finds the path with the least number of steps but ignores cost.**
3. ***A Search (A-Star)*\***
   * **Uses a heuristic (straight-line distance to cheese) to guide search.**
   * **Efficiently finds the cheapest path while exploring fewer junctions.**

**Reasons for this Approach:**

* **These are basic search algorithms we are learning in class.**
* **DFS is fast but not always optimal.**
* **BFS ensures the shortest steps.**
* **A\* ensures the lowest cost and is smart with heuristics.**

**Summary:  
By representing the pipe network as a graph and applying DFS, BFS, and A\*, we can compare the paths and costs Terry takes. This helps us see which algorithm is better for different situations.**

**5. Python Code Implementation**

**A white background with black text

AI-generated content may be incorrect.**

**A white background with black text

AI-generated content may be incorrect.**

**A white background with text

AI-generated content may be incorrect.**

**A white background with green and yellow dots

AI-generated content may be incorrect.**

**Output:**

**A white background with blue lines

AI-generated content may be incorrect.**

**6. Observations and Conclusion**

**After running BFS, DFS, and A\* algorithms, we can make the following observations:**

**1. Breadth-First Search (BFS):**

* **Explores all nearby junctions first.**
* **Finds the shortest path in terms of number of junctions but ignores cost.**
* **Explored more junctions than A\*.**

**2. Depth-First Search (DFS):**

* **Goes deep along one path before backtracking.**
* **Path may be longer and less efficient.**
* **Explores junctions in order but may not find the cheapest route.**

***3. A Search (A*):\*\***

* **Uses a heuristic to guide the search.**
* **Finds the cheapest path while exploring fewer junctions.**
* **Most efficient for helping Terry reach the cheese.**

**Overall Observations:**

* **BFS is good for shortest steps.**
* **DFS is quick but unpredictable.**
* **A\* is the best choice as it balances cost and efficiency, helping Terry reach the cheese safely and quickly.**