Name: Trevor Buck

ID: 109081318

CSCI 3104, Algorithms **Profs. Grochow & Layer**
Explain-It-Back 8 **Spring 2019, CU-Boulder**

You are collaborating with a geology team that is using a rover to explore lava fields. They have mapped out the surface of the lava field as a grid where each edge is annotated with the likelihood that the robot can successfully navigate the corresponding terrain. This likelihood integrates physical properties such as surface temperature and relief. Their current algorithm finds a path by considering all of the edges at its current position, then all of the edges that are one step away, two steps away, and so on until they reach the desired destination. Unfortunately, this process takes so much time to complete that the physical properties of the edges change before a route can be calculated rendering the path useless. You have the insight that the robot does not need the best path, just one that can be calculated quickly and has a reasonable likelihood of success. Help your team understand the issues with the current solution, and how a simple algorithm change could help.

To my favorite geology team,

I still can't believe that you were able to get funding for this project! Robots navigating through lava fields sounds like something out of a video game, and you guys are making it happen. And I have some good news for you! I heard about your problem with the robot and its navigation and I have a suggestion that should help. From what I understand, it sounds like your algorithm can find a safe path, but it takes too long and by then, the surface has changed. My suggestion is that you change your algorithm to be a 'Depth First Search' algorithm. Let me explain what this is.

To fully understand the concept of a Depth First Search, or DFS algorithm, we need to take a step back and look at the problem as a whole. There are two main types of search algorithms when it comes to finding a path. There is what we call a Depth First Search, and the other is a Breadth First Search or BFS algorithm. The algorithm that you are currently implementing is the latter. In your own words, you said that you find all the edges that are one step away, then two, then three, and so on until the destination is reached. We call this a BFS algorithm because it searches all the nearby edges before digging deeper. This is sometimes referred to as a horizontal searching algorithm. This type of algorithm explores all possible paths and is therefore able to find the shortest path. In general, the shortest path is useful to know. However, in this situation all we need is an appropriate path. This leads us to the DFS algorithm.

A DFS algorithm takes the 'run as straight and fast as you can until you hit your goal and only look at the side edges if you get stuck' approach. It takes one step, and then another, and then another, until it reaches a dead end or the goal. If we hit a dead end, then it goes back to an old vertex and looks to see if it has a different path. For those with a photographic mind, it would be called a vertical approach. This type of algorithm is useful because it finds a path as fast as it can. It may not be the optimal path because it doesn't explore every option, but it has the fastest return time. Thus it has the advantage of a quick and somewhat efficient solution, with the offset of that solution is not likely to be the optimal solution. In the situation with the lava fields, I believe that this would be helpful because you need to get that robot moving as fast as you can. So in the end, you would save time by starting the robot earlier and taking the hit on a little bit longer of a path rather than trying to find the shortest route first, and then moving.

Let me know if you have any questions. Thanks!

Trevor Buck

Name: Trevor Buck

ID: 109081318

CSCI 3104, Algorithms **Profs. Grochow & Layer**
Explain-It-Back 8 **Spring 2019, CU-Boulder**