Maze Solver

Yanhao Li (yl4734), Linyu Li (ll3465)

CS4995 FInal Project Proposal
November 22th, 2021

1. Introduction

The maze is a path or collection of paths, typically from an entrance to a goal. In our project, we will set the left upper corner as the starting point of the maze, and set the right lower corner as the ending point of the maze. And we will also set “0” as the passable path, “1” as the block in the path which can not be passed.

Maze solving is the act of finding a route through the maze from the start to finish. Some maze solving methods are designed to be used inside the maze by a traveler with no prior knowledge of the maze, whereas others are designed to be used by a person or computer program that can see the whole maze at once.

Mazes containing no loops are known as "standard", or "perfect" mazes, and are equivalent to a tree in graph theory. Thus many maze solving algorithms are closely related to graph theory. Intuitively, if one pulled and stretched out the paths in the maze in the proper way, the result could be made to resemble a tree.

2. Algorithms

We can either use BFS (Breadth-First-Algorithm) or backtracking:
The breadth-first-search algorithm is a way to explore the vertices of a graph layer by layer. It is a basic algorithm in graph theory which can be used as a part of other graph algorithms.

```
1    define bfs_sequential(graph(V,E), source s):
2        for all v in V do
3            d[v] = -1;
4        d[s] = 0; level = 0; FS = {}; NS = {};
5        push(s, FS);
6        while FS !empty do
7            for u in FS do
8                for each neighbour v of u do
9                    if d[v] = -1 then
10                       push(v, NS);
11                       d[v] = level;
12           FS = NS, NS = {}, level = level + 1;
```

3. Parallelization

To optimize the algorithm with parallelization, we will use the following two potential approaches:

a. Split the maze into several matrices, each part will be solved individually, the destination of a single matrix will be the start point of its consecutive matrix.
b. Go down the path and when we have multiple choices at a single point, delegate each potential path to a single thread, as there are at most three directions to go in the next step. However, for this approach, we will have to deal with the synchronization issues.

4. Objective and Deliverable

Objective: The goal of this project is to generate a path between a start and end in a maze game. This code will play the maze game. And we plan to use parallel programming to improve the efficiency of the maze solving process so as to reduce the solving time of traditional serial maze solver algorithms.

Deliverable: For the completion of this project, we will provide serial and parallelized implementations of the maze solver. Additionally, we will provide test results showing comparisons of the two versions of the algorithm.