CSE12 - Lecture 8

Wednesday, April 19, 2023 8:00 AM

PA2 → tonight @ 10pm

PA3 → released → due Tuesday

2D Search

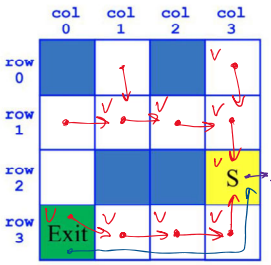
Breadth-First Search (BFS)

Guaranteed shortest path

```
class Square {
  boolean visited;
  Square previous;
}
```

class Node { boolean visited; Node next; }

S



Null

SearchForTheExit

```
Initialize a Queue to hold Squares as we search
Mark starting square as visited
Enqueue starting square on Queue
While Queue is not empty
    Dequeue square sq from Queue
    Mark sq as visited
    If sq is the Exit, we're done!
    For each of square's unvisited neighbors (S, W, N, E):
        Set neighbor's previous to sq
        Enqueue neighbor to Queue
```

Run through the SearchForTheExit algorithm. Draw the queue.

front → (2,3) (3,3) (1,3) (3,2) (1,2) (0,3) (3,1) (1,1) (3,0) (1,0) (0,1)

Exit (3,0) → (3,1) → (3,2) → (3,3) → (2,3) → Null

start

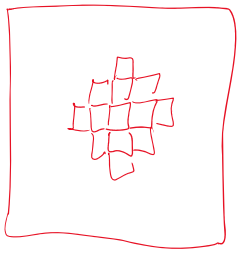
use a stack to reverse

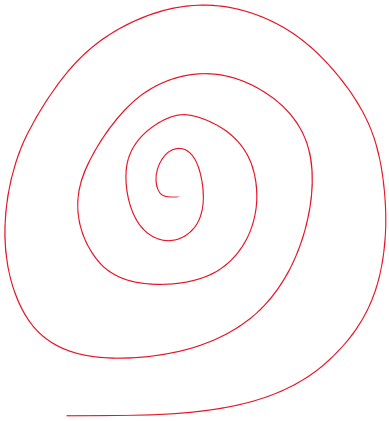
How many nodes were visited? 9

How many total squares were added to the queue? 11

Was this the shortest path? yes

Name: _____ PID: _____ Code: 1668

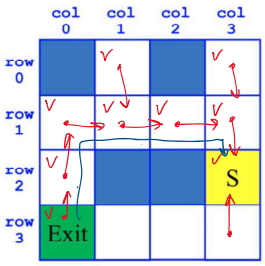




Depth-First Search (DFS)

Will always find a path. Possibly faster.

```
class Square {
  boolean visited;
  Square previous;
}
```



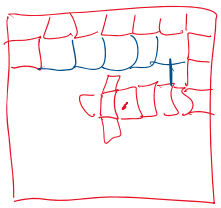
SearchForTheExit

```
Initialize a Stack to hold Squares as we search
Mark starting square as visited
Push starting square on Stack
While Stack is not empty
    Pop square sq from Stack
    Mark sq as visited
    If sq is the Exit, we're done!
    For each of square's unvisited neighbors (S, W, N, E):
        Set neighbor's previous to sq
        Push neighbor to Stack
```

4 3 2 1

Run through the SearchForTheExit algorithm. Draw the stack.

bottom → (2,3) (3,3) (1,3) (1,2) (0,3) (1,1) (1,0) (0,1)
(2,0) (3,0)



How many nodes were visited? 9

How many total squares were added to the stack? 10

Was this the shortest path? NO

A* → A star