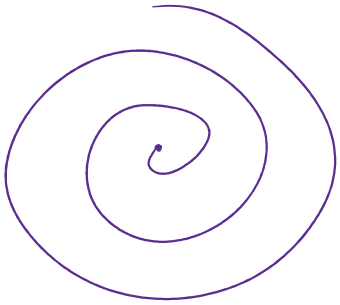
CSE12 - Lecture 8

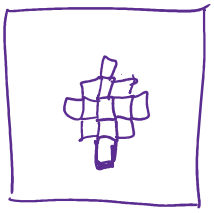
Wednesday, April 17, 2024 10:00 AM

PA3 released - due next Wednesday @ 8am
PA2 - hard deadline - Thursday @ 8am

Exam 1 - next Wednesday

- https://ucsd-cse12-sp24.github.io/lectures/exam1.html





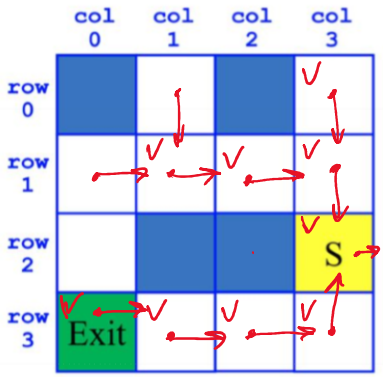
Lecture 8

Breadth-First Search (BFS)

Guaranteed shortest path

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

class Node
  boolean visited;
  Node next;
;




```
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
```

null

S → (2,3)

S W N E
1 2 3 4

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
5 squares
start

use stack to reverse

How many nodes were visited? 9

How many total squares were added to the queue? 11

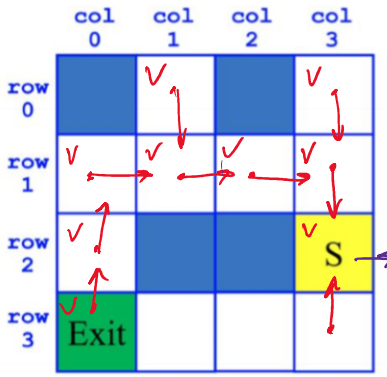
Was this the shortest path? yes

Name: _____ PID: _____ Code: 2392

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
```

null

S W N E
4 3 2 1

Run through the SearchForTheExit algorithm. Draw the stack.

bottom → ~~(3,3)~~ (3,3) ~~(1,3)~~ ~~(1,2)~~ ~~(0,3)~~ ~~(1,1)~~ ~~(1,0)~~ ~~(0,1)~~ ← top

~~(2,0)~~ ~~(3,0)~~

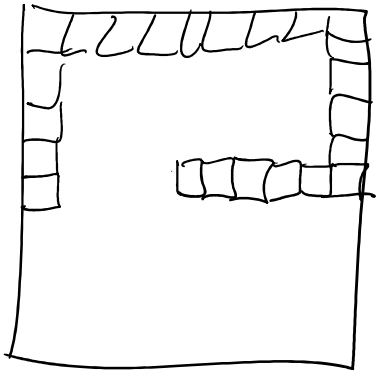
exit → (3,0) → (2,0) → (1,0) → (1,1) → (1,2) → (1,3) → (2,3) → null

7 squares

How many nodes were visited? 9

How many total squares were added to the stack? 10

Was this the shortest path? No



Dijkstra's

A*