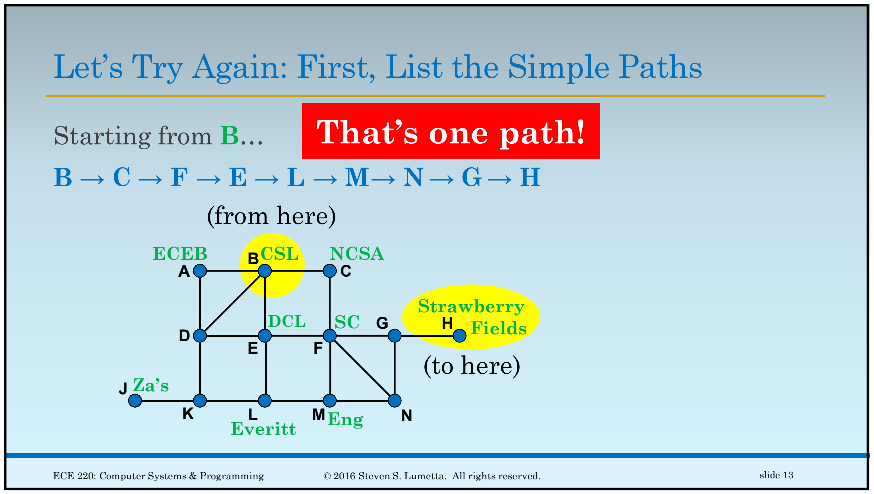


Let's Try Again: First, List the Simple Paths

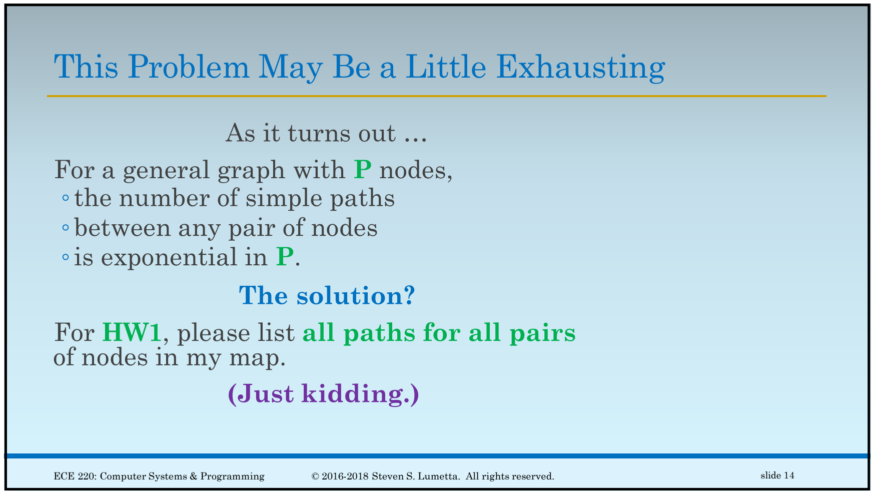
Starting from **B**… **That's one path!**

B → C → F → E → L → M→ N → G → H

(from here)

ECEB
A
B
CSL
NCSA
C
DCL
SC
G
Strawberry
H
Fields
D
E
F
J
Za's
K
L
Everitt
M
Eng
N

(to here)



This Problem May Be a Little Exhausting

As it turns out …

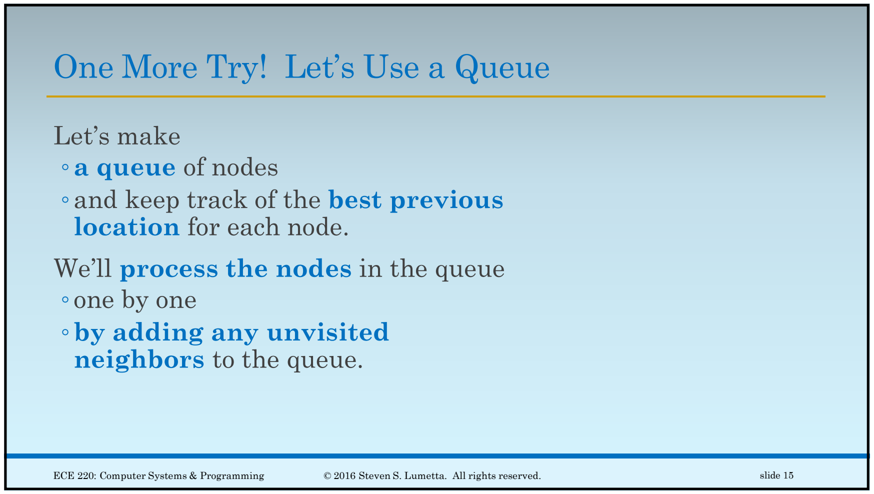
For a general graph with **P** nodes,

- the number of simple paths
- between any pair of nodes
- is exponential in **P**.

The solution?

For **HW1**, please list **all paths for all pairs** of nodes in my map.

(Just kidding.)



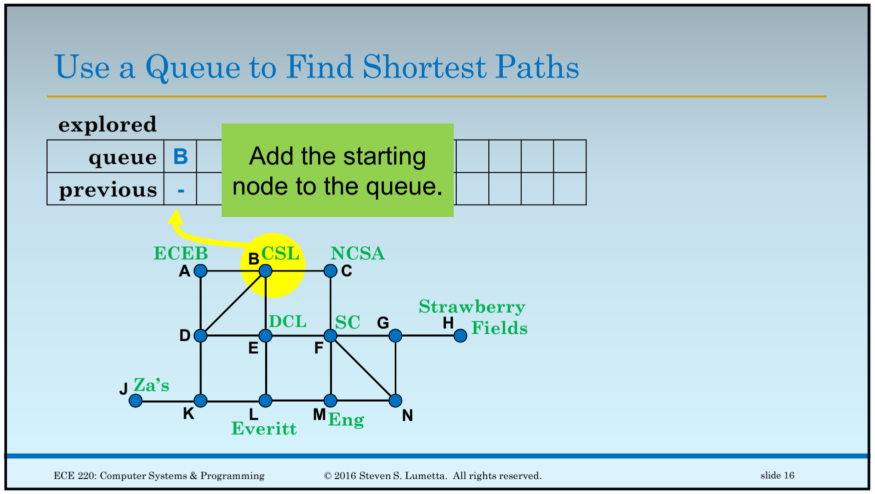
One More Try! Let's Use a Queue

Let's make

- **a queue** of nodes
- and keep track of the **best previous location** for each node.

We'll **process the nodes** in the queue

- one by one
- **by adding any unvisited neighbors** to the queue.



Use a Queue to Find Shortest Paths

explored

queue	B	
previous	-	

Add the starting node to the queue.

ECEB
A
B
CSL
NCSA
C
DCL
SC
G
Strawberry
H
Fields
D
E
F
J
Za's
K
L
Everitt
M
Eng
N