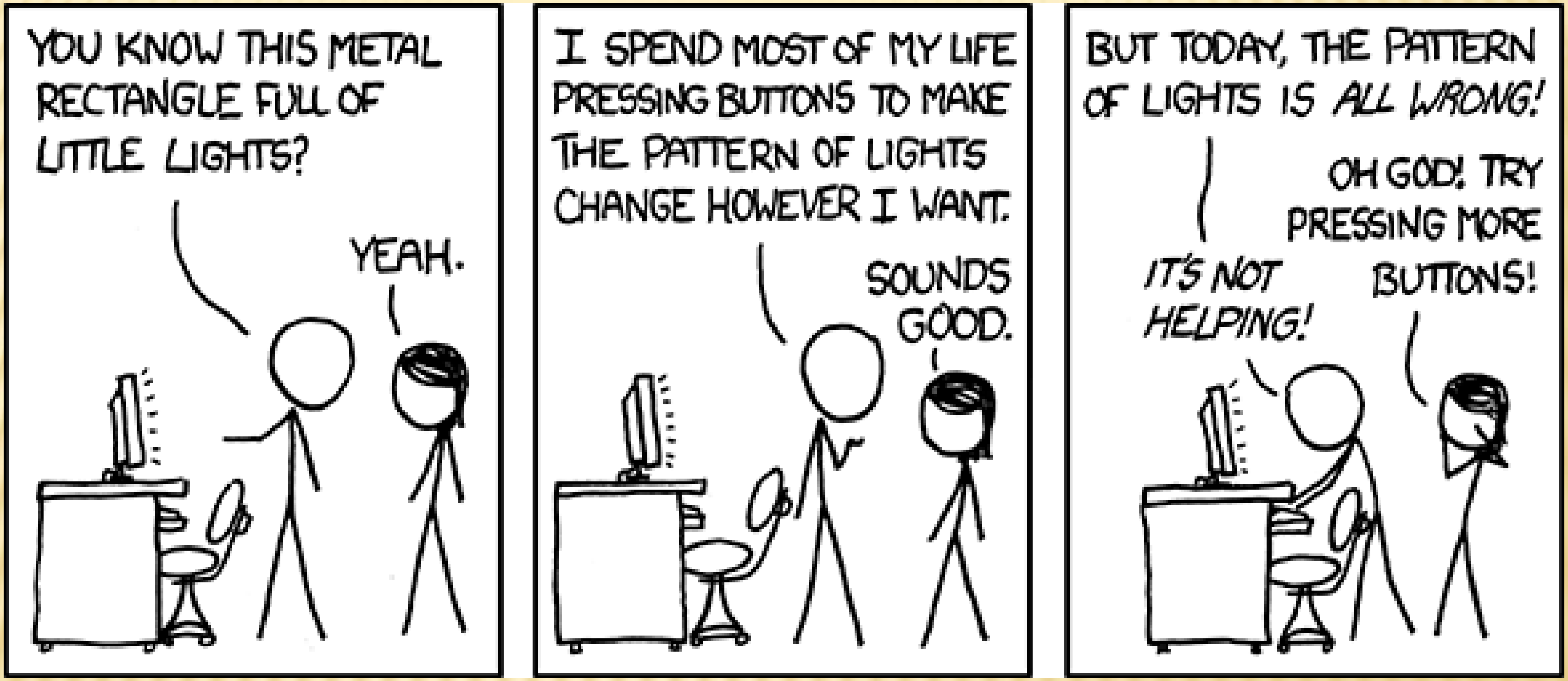


YOU KNOW THIS METAL RECTANGLE FULL OF LITTLE LIGHTS?
YEAH.
I SPEND MOST OF MY LIFE PRESSING BUTTONS TO MAKE THE PATTERN OF LIGHTS CHANGE HOWEVER I WANT.
SOUNDS GOOD.
BUT TODAY, THE PATTERN OF LIGHTS IS *ALL WRONG!*
OH GOD! TRY PRESSING MORE BUTTONS!
IT'S NOT HELPING!

Section 6:

HW6

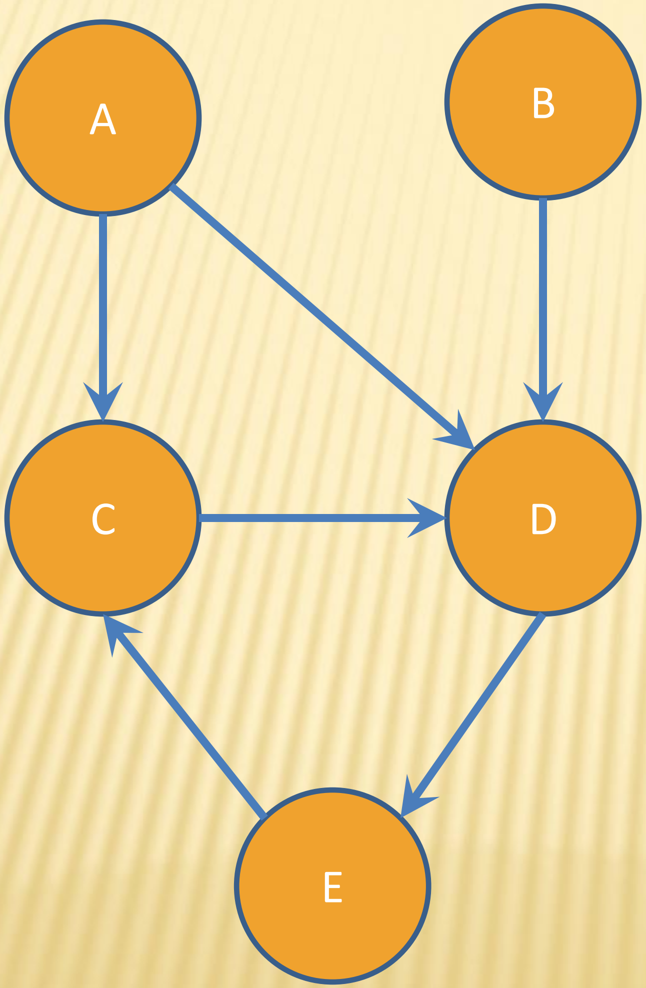
Slides by Vinod Rathnam

with material from Alex Mariakakis, Krysta Yousoufian, Mike Ernst, Kellen Donohue

HANDLING EXPENSIVE RIS

- **Problem:** a thorough `checkRep()` may take a while to execute; if it is called every time the graph is modified, your code may fail the 30 second timeout per test
- **Simple solution:** use a “debug flag” boolean to turn `checkRep()` on or off **(Do this!)**
- **Fancy solution:** make multiple `checkRep()` methods of different complexity and switch between them using an enum

GRAPHS



Can I reach B from A?

BREADTH-FIRST SEARCH (BFS)

- Often used for discovering connectivity
- Calculates the shortest path if and only if all edges have same positive or no weight
- Depth-first search (DFS) is commonly mentioned with BFS
  - BFS looks “wide”, DFS looks “deep”
  - Can also be used for discovery, but not the shortest path

BFS PSEUDOCODE

```
public boolean find(Node start, Node end) {
      put start node in a queue
      while (queue is not empty) {
            pop node N off queue
            if (N is goal)
                  return true;
            else {
                  for each node O that is child of N
                        push O onto queue
            }
      }
      return false;
}
```

BREADTH-FIRST SEARCH

Q: <>

Q: <A>

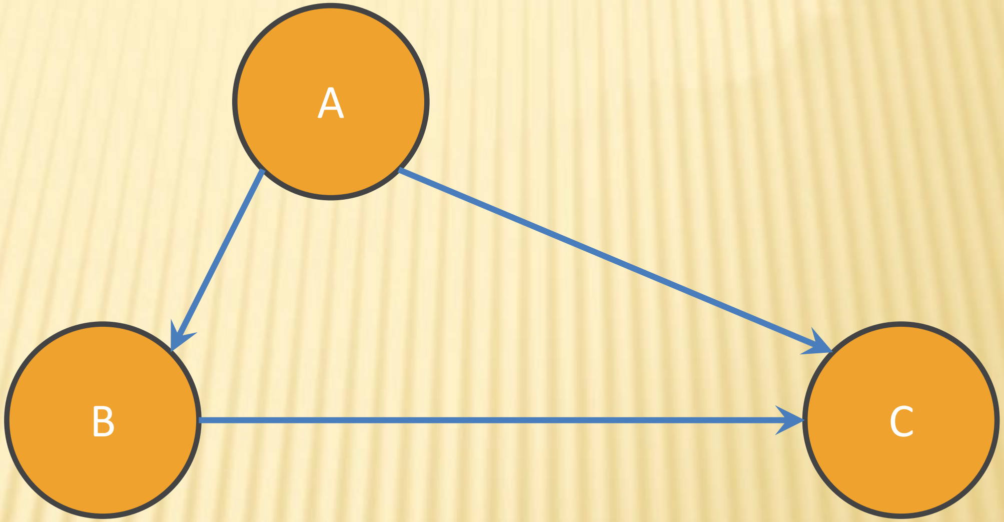
Q: <>

Q: <B>

Q: <B, C>

DONE

Starting at node A



BREADTH-FIRST SEARCH WITH CYCLE

Q: <>

Q: <A>

Q: <>

Q: <B>

Q: <>

Q: <C>

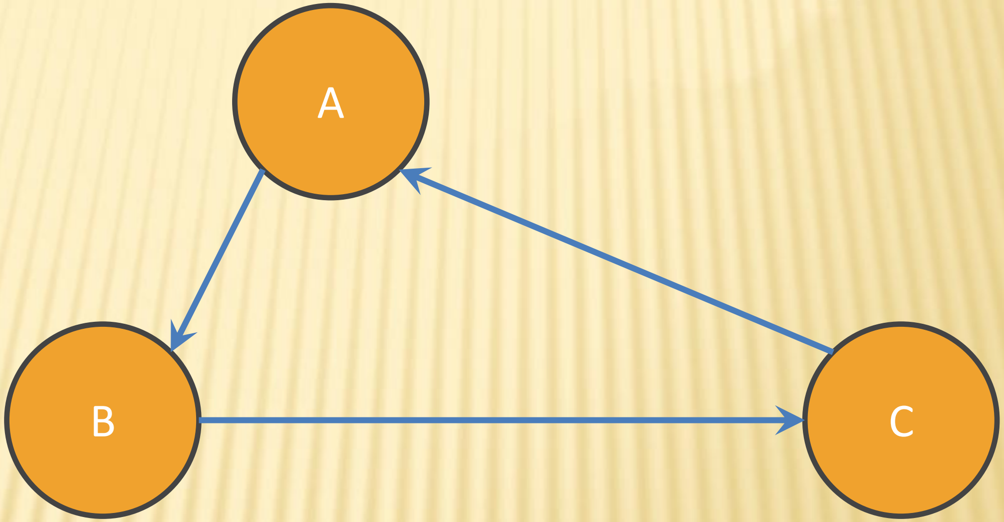
Q: <>

Q: <A>

NEVER

DONE

Starting at node A



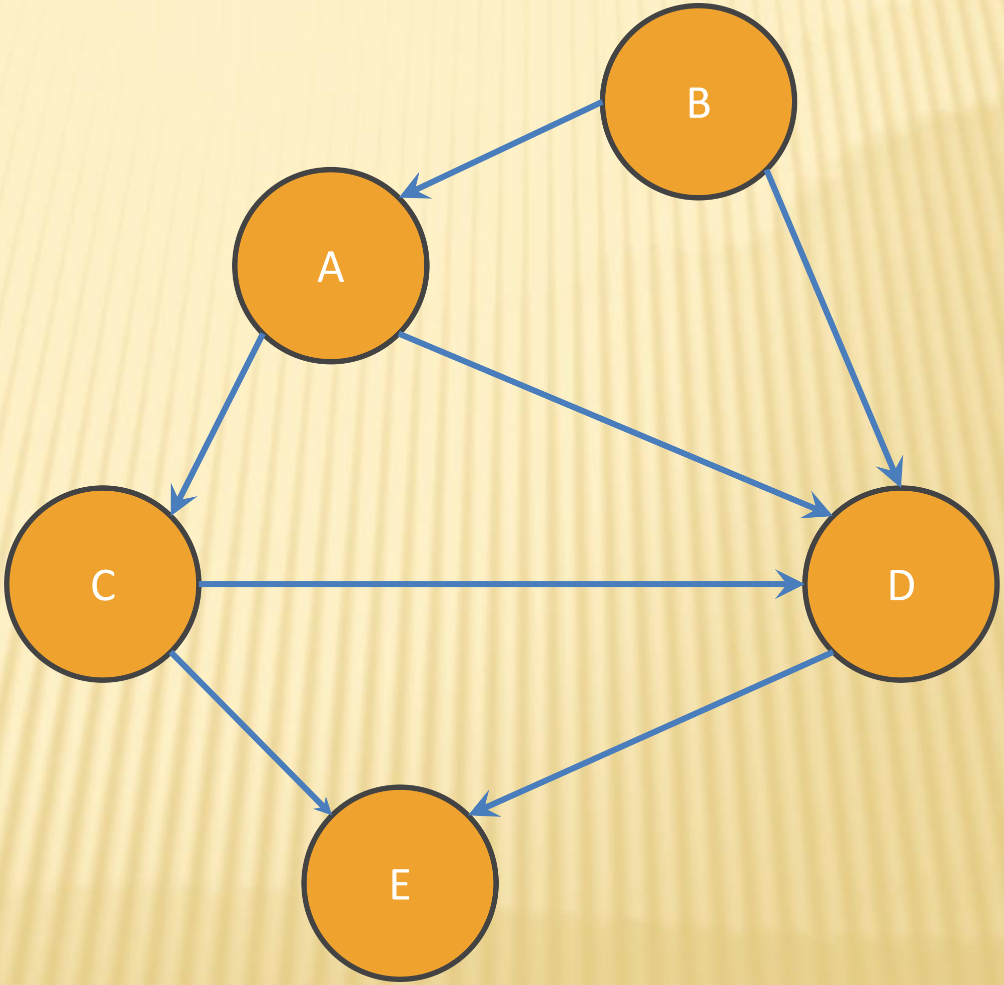
BFS PSEUDOCODE

```
public boolean find(Node start, Node end) {
    put start node in a queue
    while (queue is not empty) {
        pop node N off queue
        if (N is goal)
            return true;
        else {
            for each node O that is child of N
                push O onto queue
        }
    }
    return false;
}
```

Mark the node as visited!

BREADTH-FIRST SEARCH

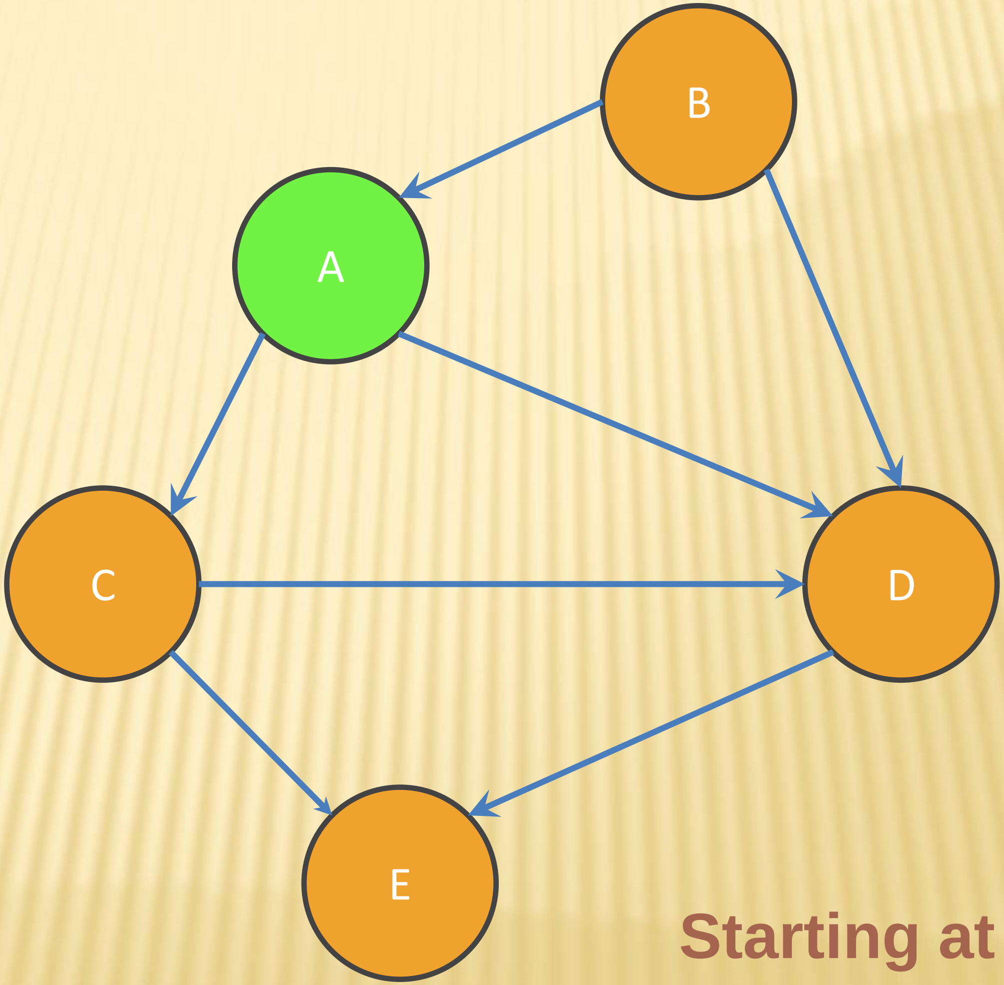
Q: <>



BREADTH-FIRST SEARCH

Q: <>

Q: <A>



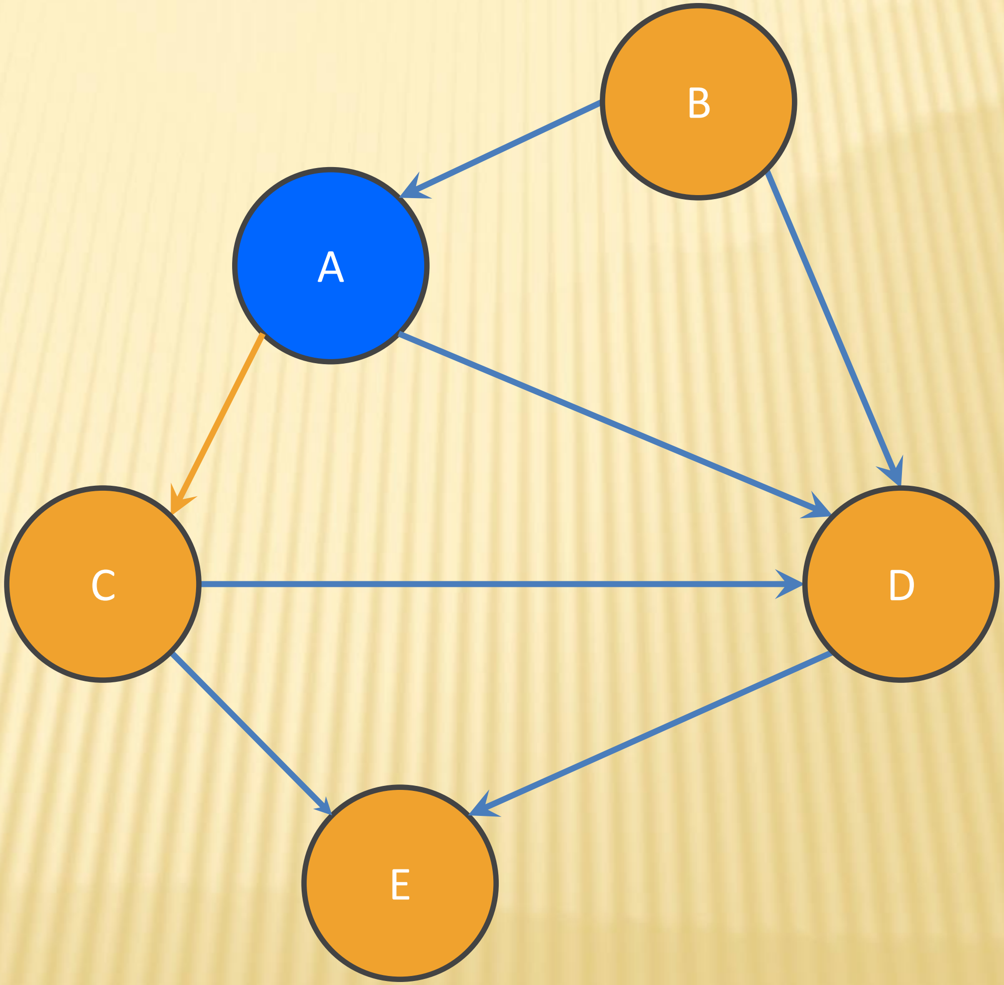
Starting at node A
Destination node is E

BREADTH-FIRST SEARCH

Q: <>

Q: <A>

Q: <>



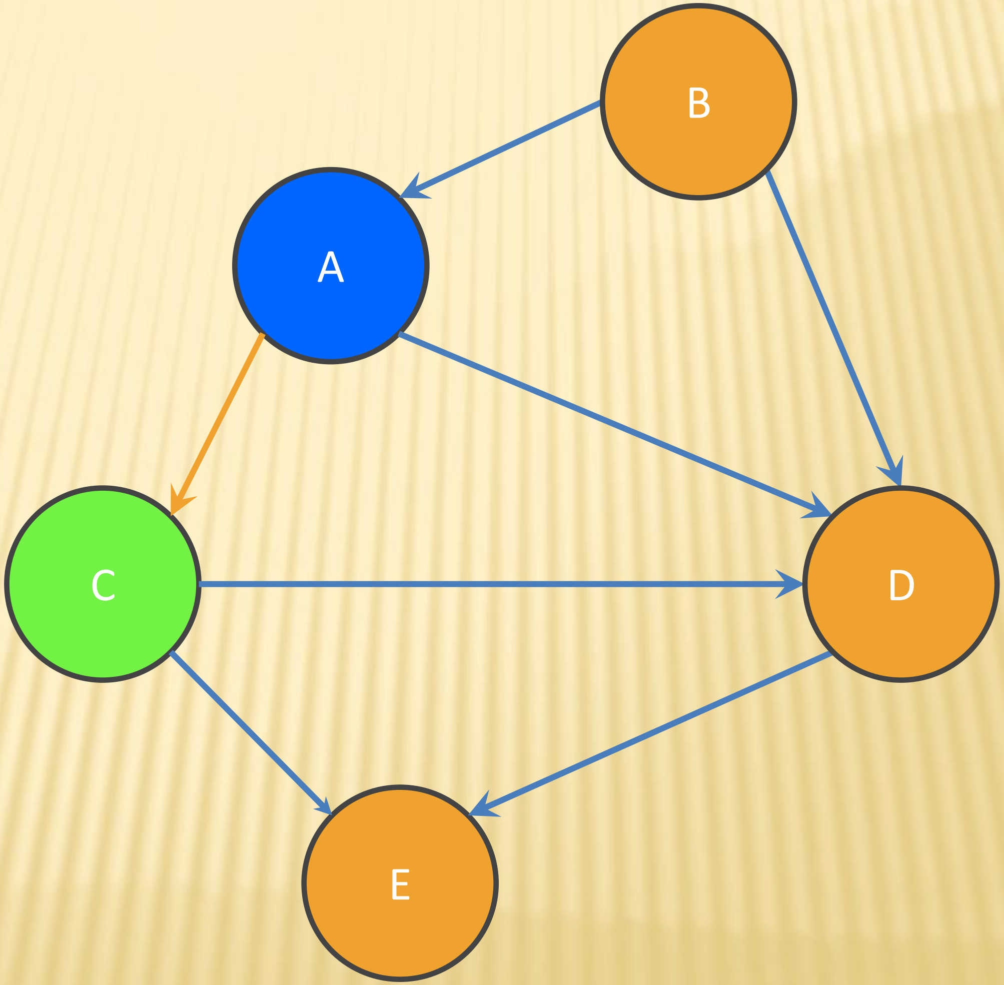
BREADTH-FIRST SEARCH

Q: <>

Q: <A>

Q: <>

Q: <C>



BREADTH-FIRST SEARCH

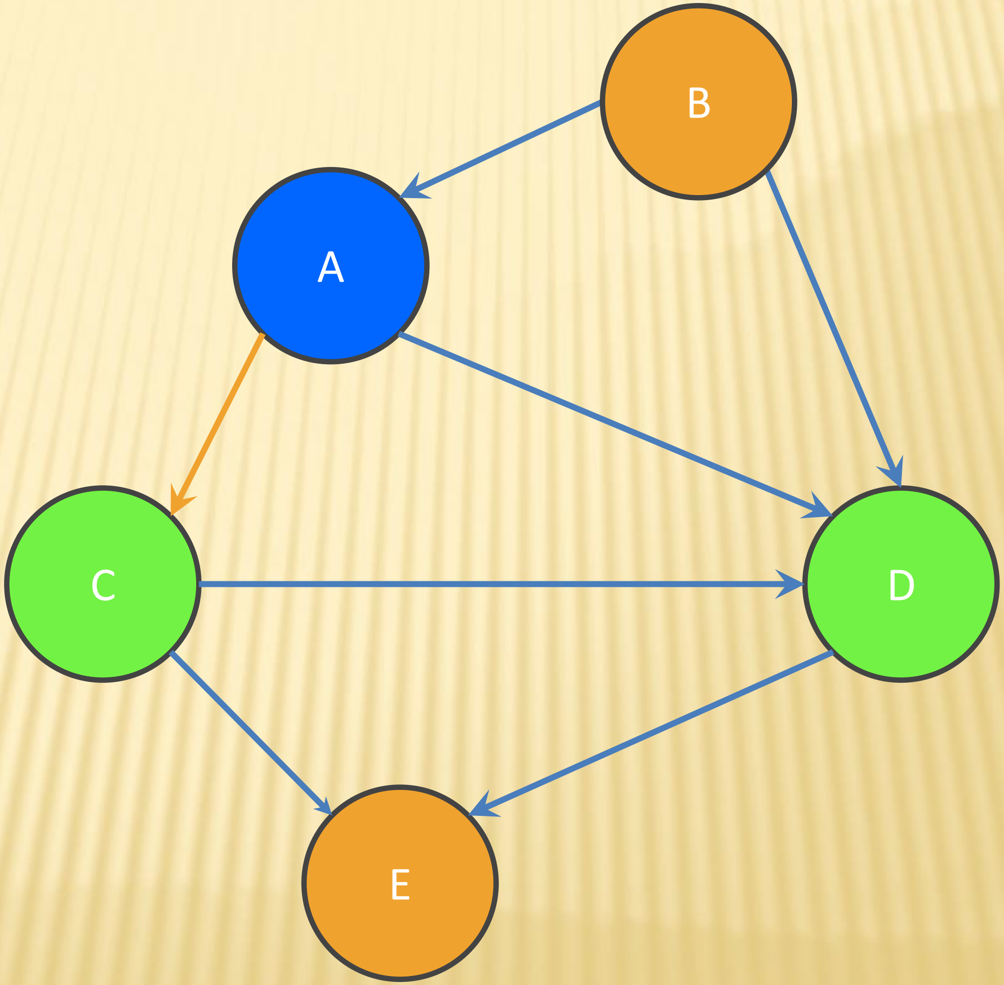
Q: <>

Q: <A>

Q: <>

Q: <C>

Q: <C ,D>



BREADTH-FIRST SEARCH

Q: <>

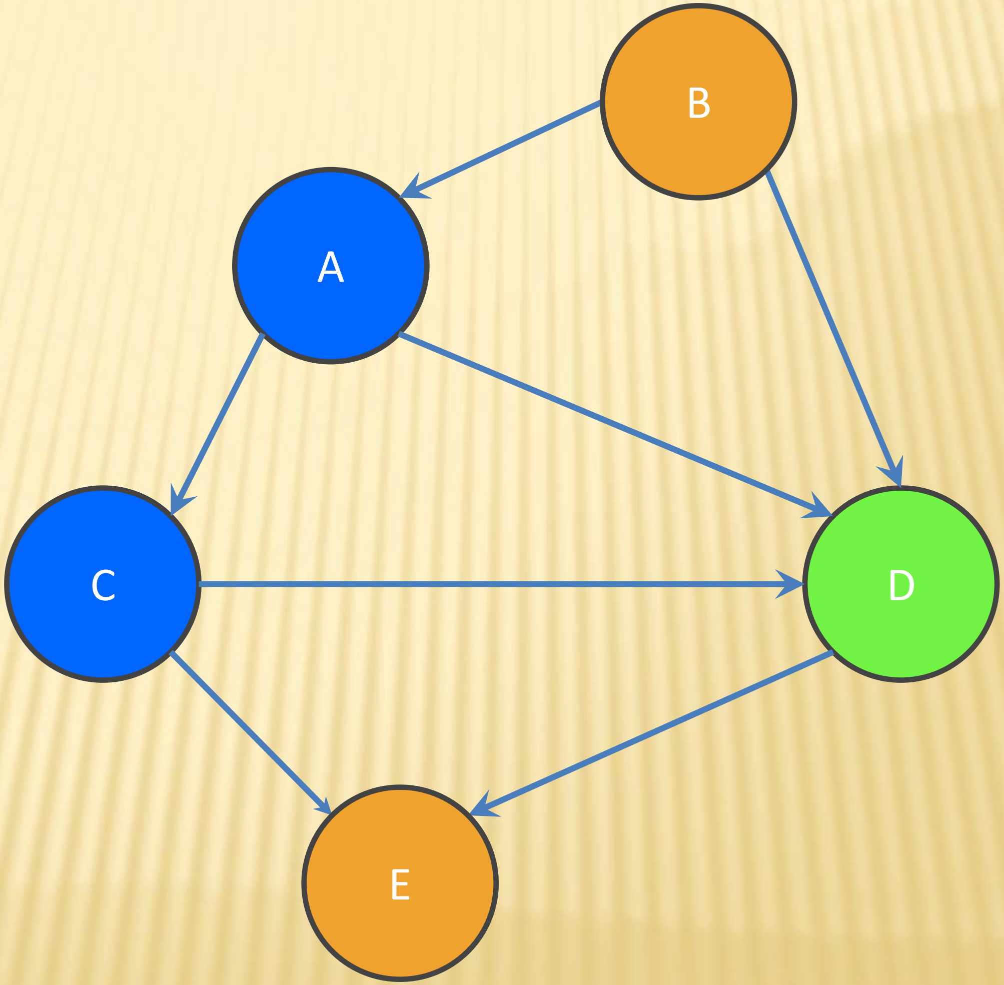
Q: <A>

Q: <>

Q: <C>

Q: <C ,D>

Q: <D>



BREADTH-FIRST SEARCH

Q: <>

Q: <A>

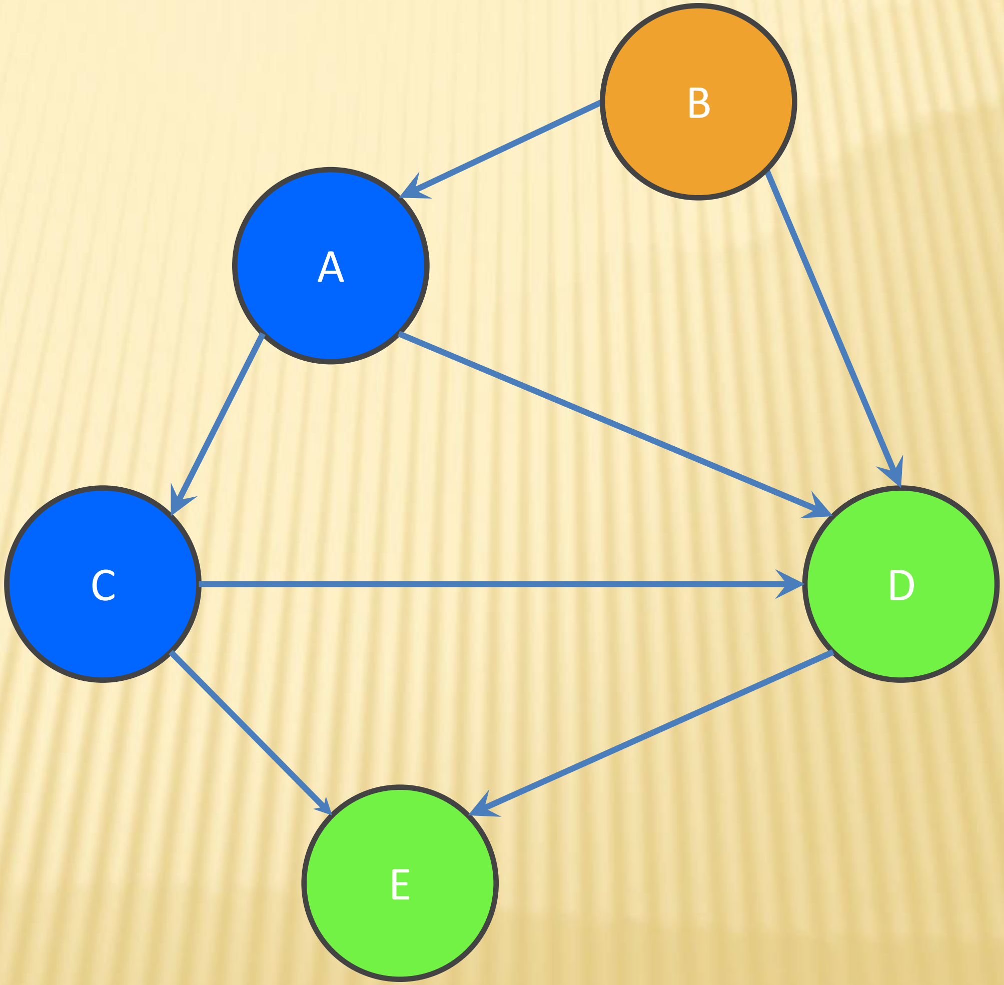
Q: <>

Q: <C>

Q: <C ,D>

Q: <D>

Q: <D, E>



Breadth-first search

Q: <>

Q: <A>

Q: <>

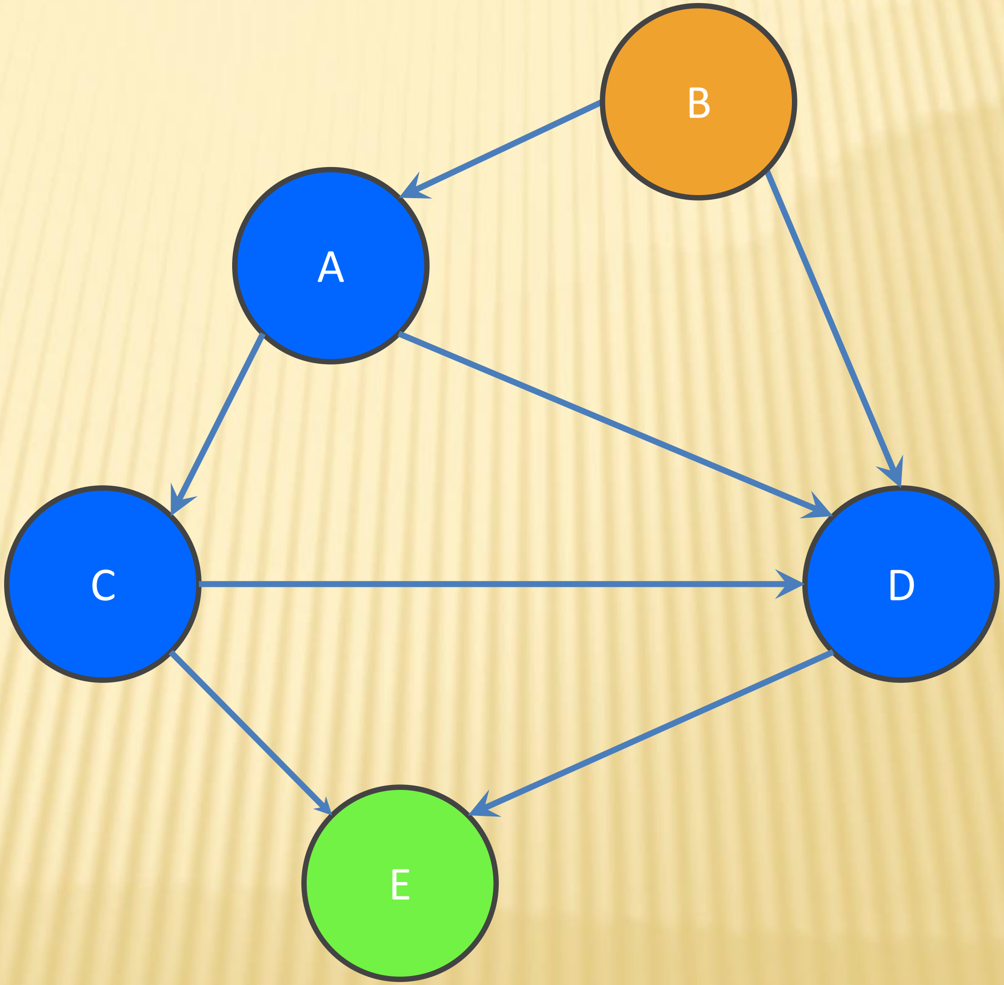
Q: <C>

Q: <C ,D>

Q: <D>

Q: <D, E>

Q: <E>



BREADTH-FIRST SEARCH

Q: <>

Q: <A>

Q: <>

Q: <C>

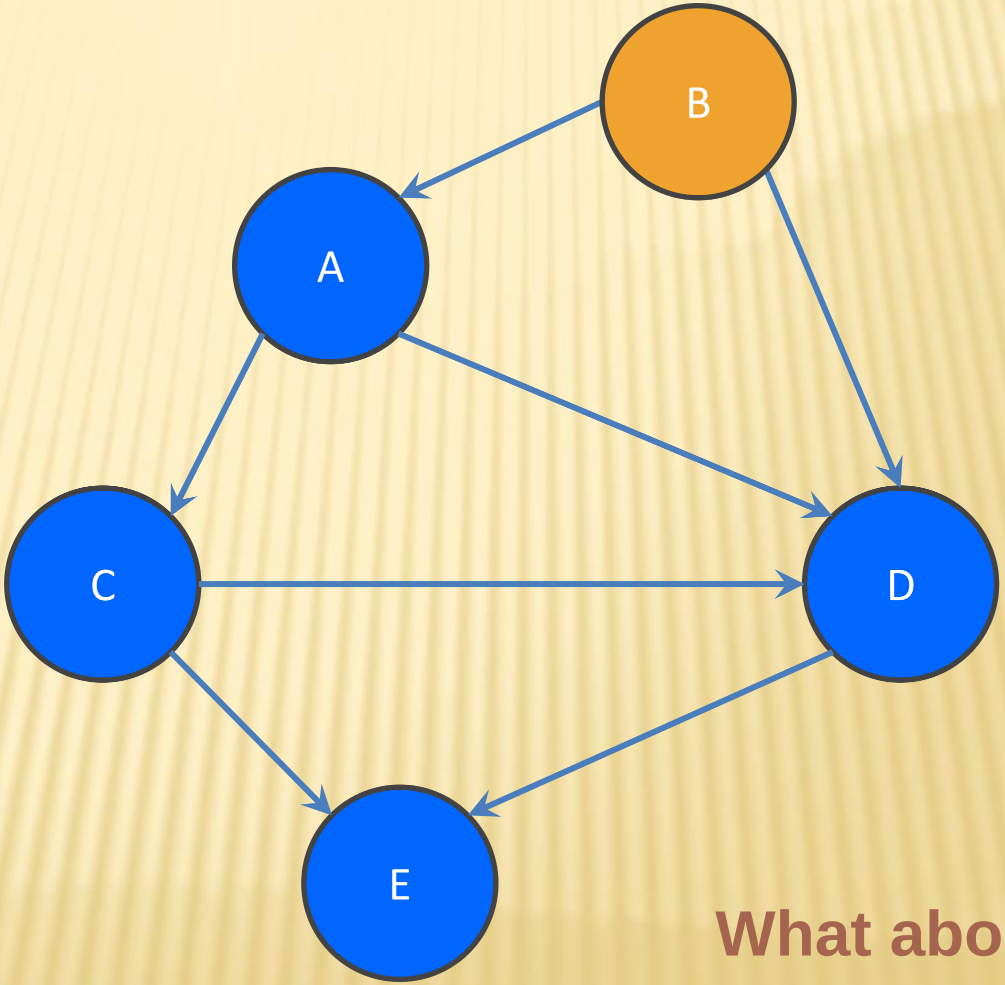
Q: <C ,D>

Q: <D>

Q: <D, E>

Q: <E>

DONE



What about B?

BREADTH-FIRST SEARCH

Q: <>

Q: <A>

Q: <>

Q: <C>

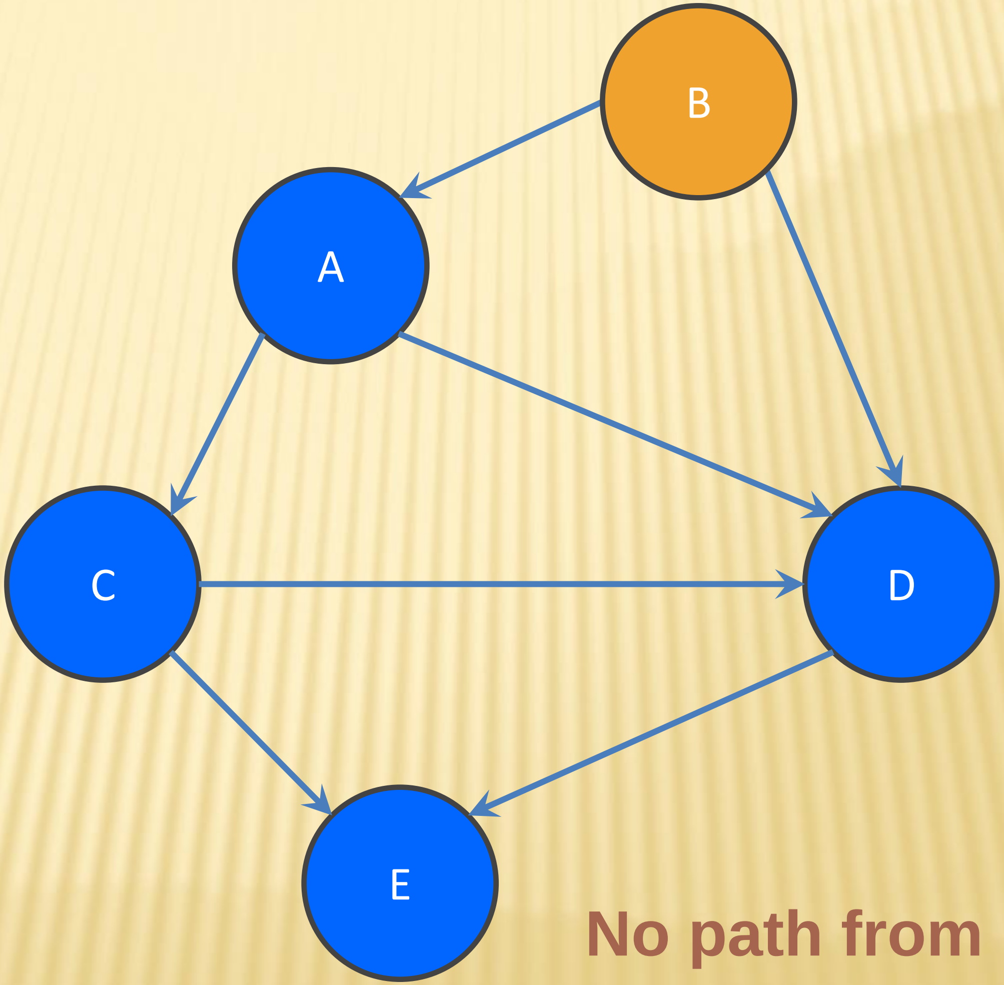
Q: <C ,D>

Q: <D>

Q: <D, E>

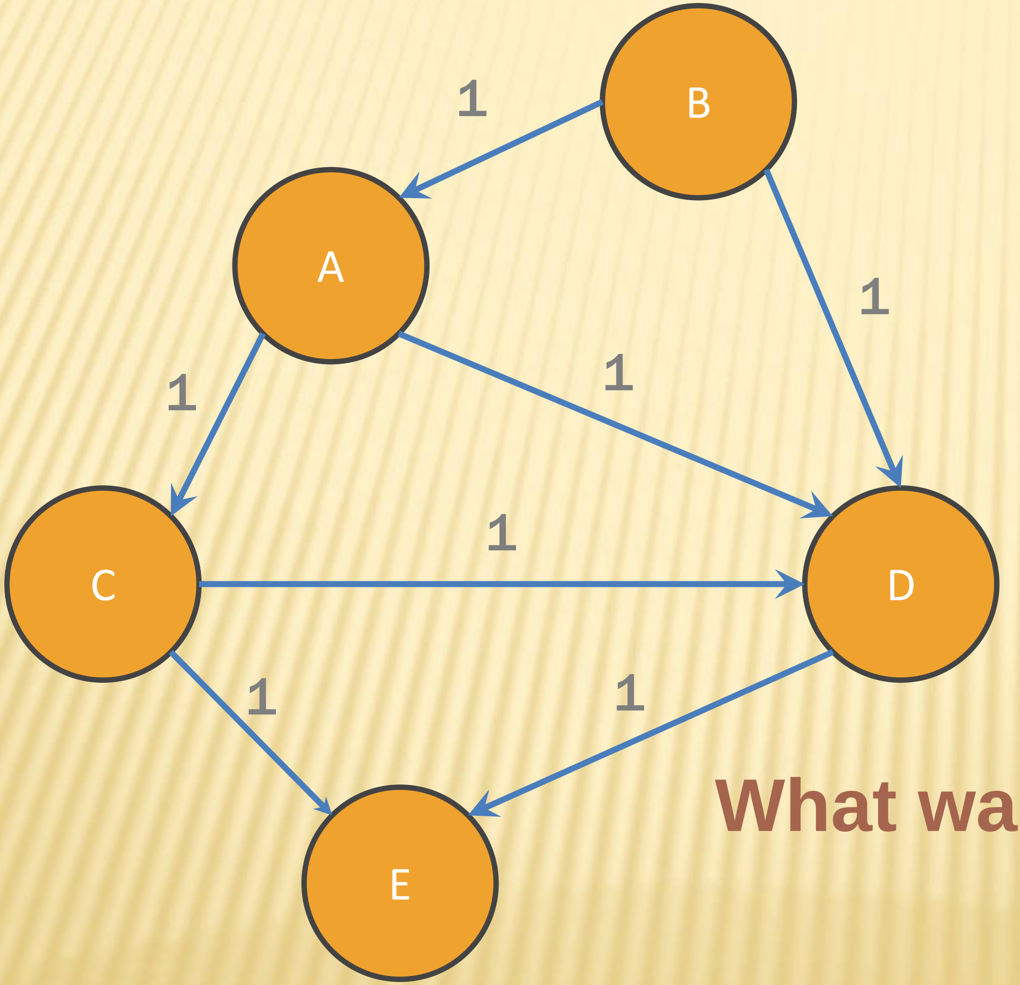
Q: <E>

DONE



No path from B to A, so not included

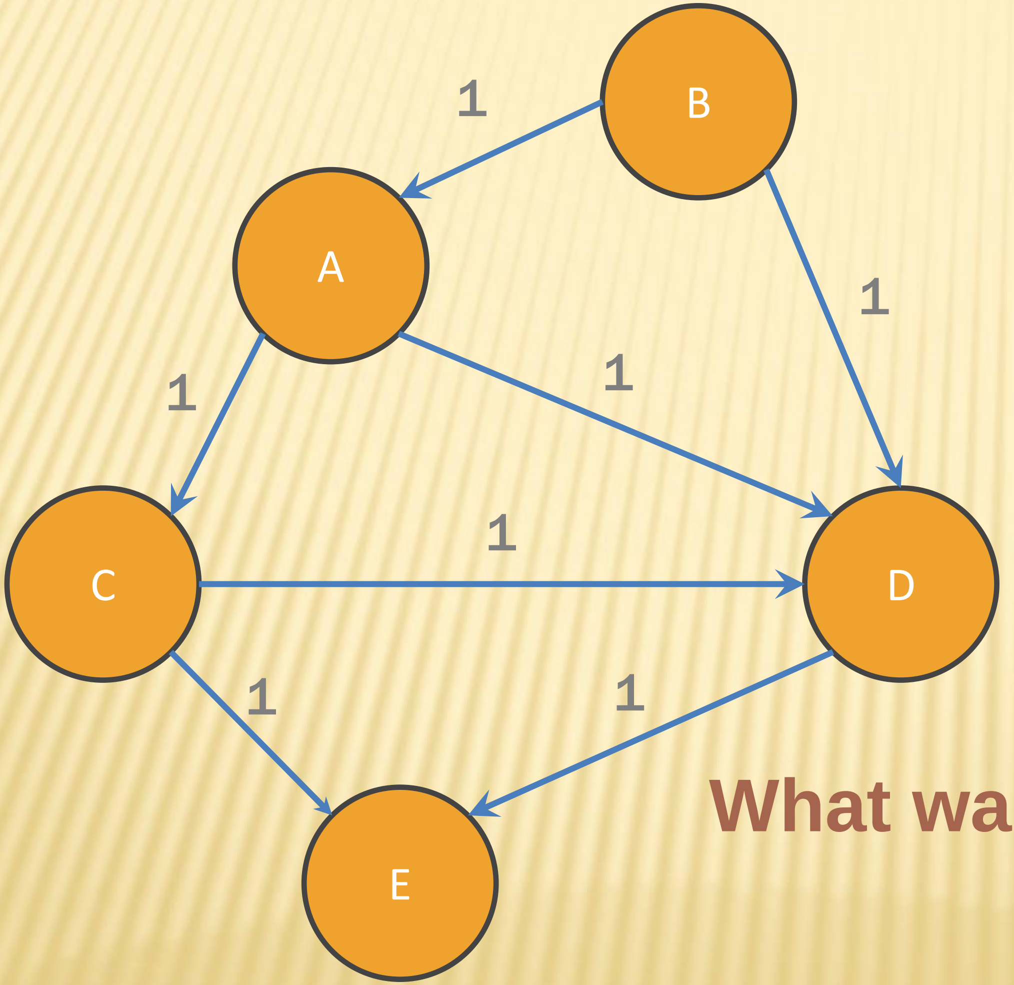
SHORTEST PATHS WITH BFS



Destination	Path	Cost
A	<B,A>	1
B	<B>	0
C	<B,A,C>	2
D	<B,D>	1
E	<B,D,E>	2

What was the starting node?

SHORTEST PATHS WITH BFS

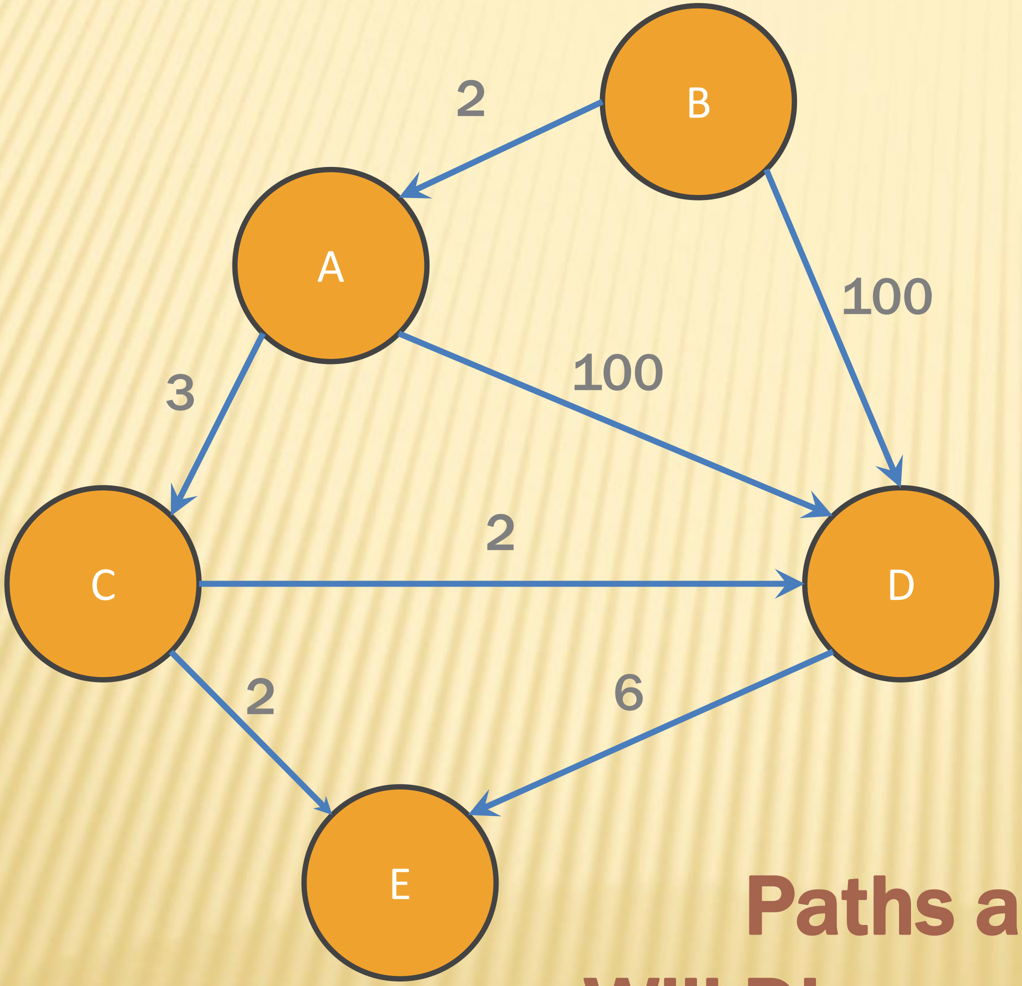


Starting From Node B

Destination	Path	Cost
A	<B,A>	1
B	<B>	0
C	<B,A,C>	2
D	<B,D>	1
E	<B,D,E>	2

What was the starting node?
Node B

SHORTEST PATHS WITH WEIGHTS



Starting From Node B

Destination	Path	Cost
A	<B,A>	2
B	<B>	0
C	<B,A,C>	5
D	<B,A,C,D>	7
E	<B,A,C,E>	7

Paths are not the same!

Will Discuss Next Week for HW7!

HW6 OVERVIEW

- Look at marvel.tsv file
- Parsing of file done for you, look at MarvelParser.java
- Fill up your graph (may need to make changes to your Graph ADT)
- Find shortest path between two characters through the different comic books that they appear in with other characters
- Testing!

Demo

HW6