

CSCI 136 Data Structures & Advanced Programming

Jeannie Albrecht
Lecture 33
~~May 9, 2014~~
May 12, 2014

Administrative Details

- Lab 10 – due today at noon
 - Any questions?
- This week
 - Wrap up graphs on Wed
 - Hashmaps on Wed and Fri
 - Course evals on Fri
- Optional lab this week
- MAKEUP CLASS WED 1:10-2:00 in Wege
- Self-scheduled final
- I will be out of town Friday - Sunday

2

Last Time

- Looked at GraphMatrix and GraphList implementations
 - Adjacency Matrix
 - Define graph using an adjacency matrix
 - Matrix keeps track of edge weights
 - Adjacency List
 - Each vertex keeps track of out edges

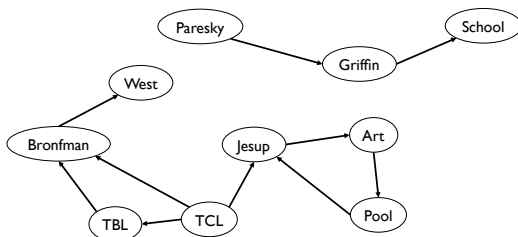
3

Today's Outline

- Continue talking about graphs
- Darwin!!

4

Suppose we applied a direction to the edges in our campus map...



5

Reachability

- There are two ways to measure reachability in our graph
 - Depth-first search and breadth-first search
- How did we do DFS and BFS in trees?
- DFS uses a stack
 - Stack records path from src to current node
 - Like pre-order tree traversal with visited flags to only visit nodes once
 - Runtime: $O(|E|)$
- BFS uses a queue
 - Queue records nodes whose out edges have not been explored
 - Like level-order tree traversal
 - Runtime: $O(|E|)$

6

Depth-First Search

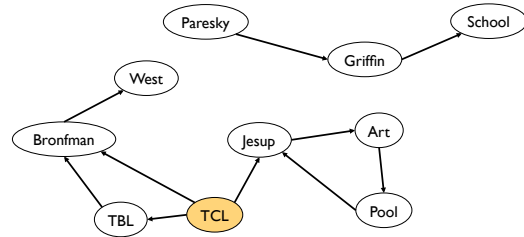
```

void reachableFrom(Graph<V,E> g, V src) {
    if (!g.visited(src)) {
        g.visit(src);

        Iterator<V> neighbors = g.neighbors(src);
        while (neighbors.hasNext()) {
            V next = neighbors.next();
            if (!g.visited(next))
                reachableFrom(g, next);
        }
    }
}
    
```

7

Depth-First Search

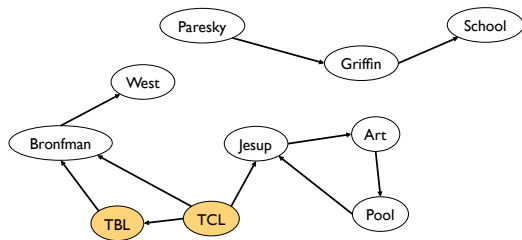


Find first unvisited neighbor of TCL...

TCL

todo stack 8

Depth-First Search

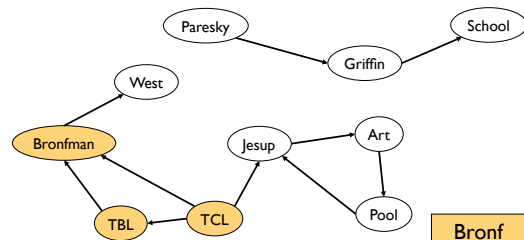


Find first neighbor of TBL...

TBL
TCL

todo stack 9

Depth-First Search

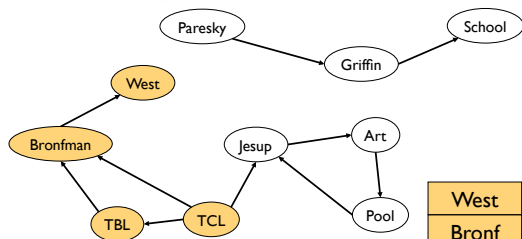


Find first neighbor of Bronfman...

Bronf
TBL
TCL

todo stack 10

Depth-First Search

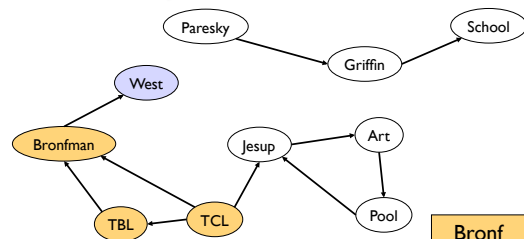


Find first neighbor of West...none exist! Pop stack...

West
Bronf
TBL
TCL

todo stack 11

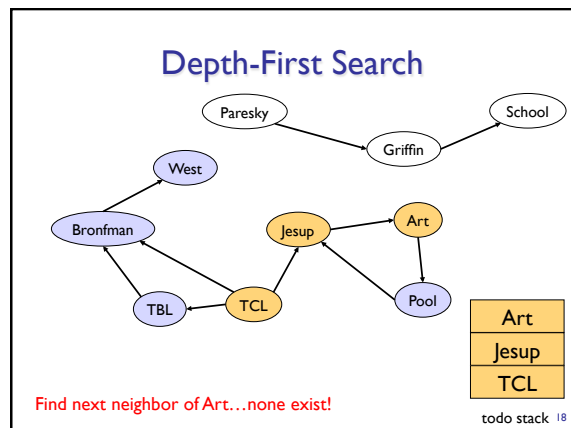
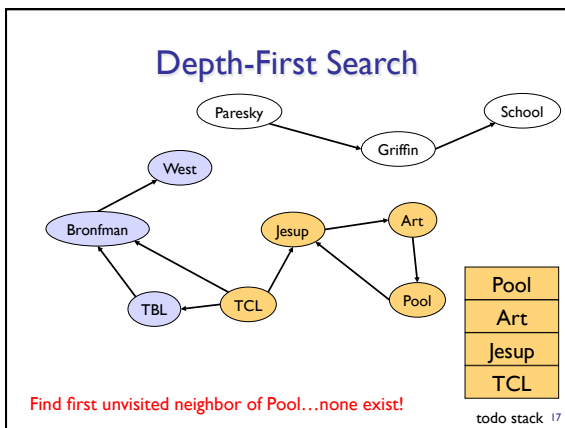
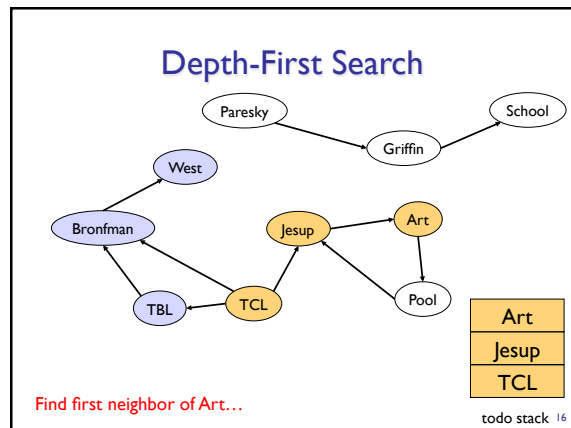
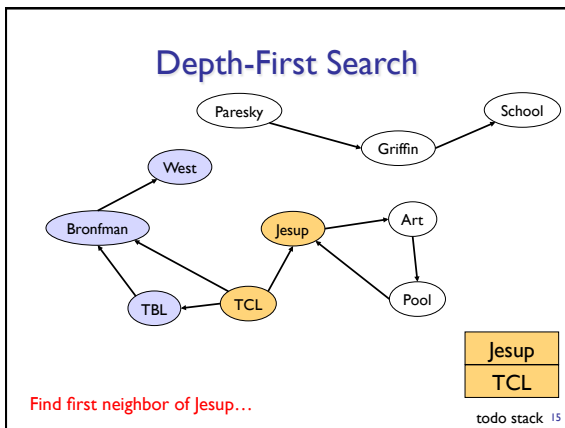
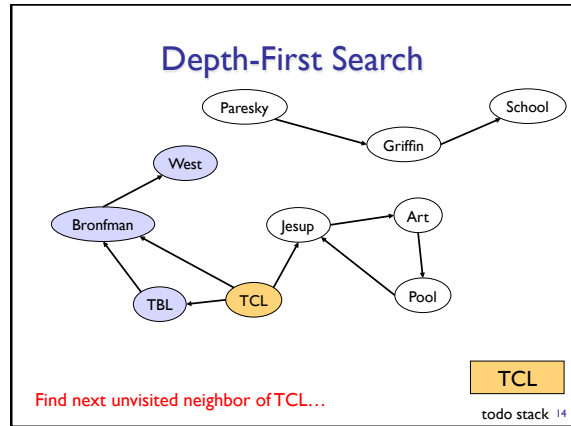
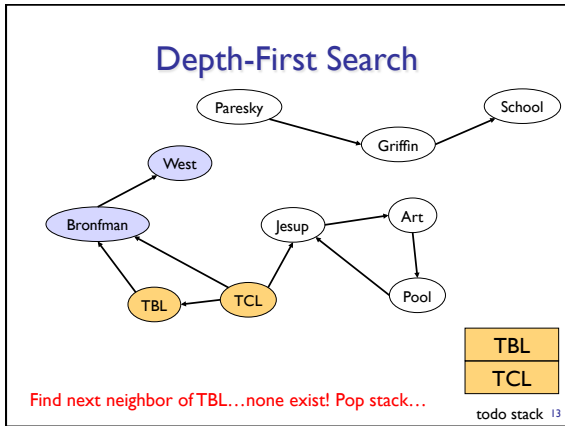
Depth-First Search

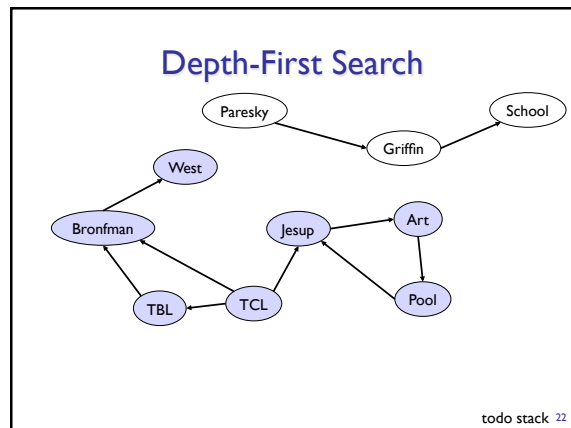
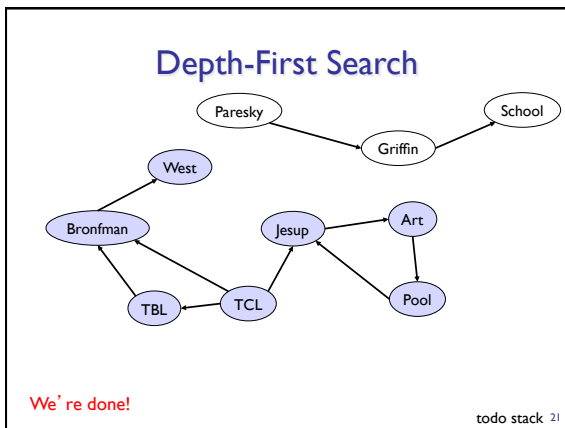
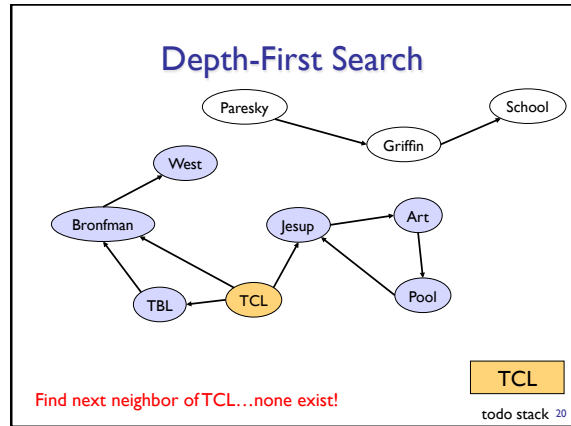
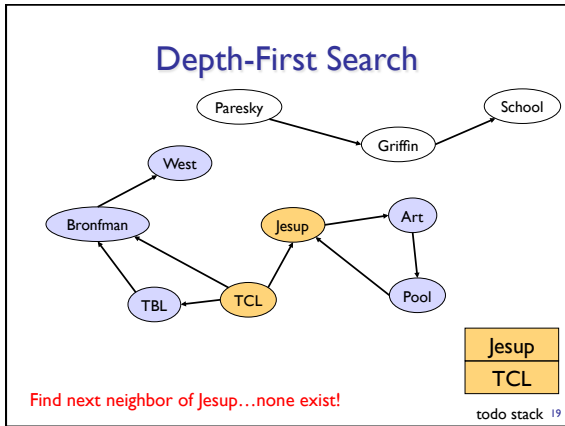


Find next neighbor of Bronf...none exist! Pop stack...

Bronf
TBL
TCL

todo stack 12



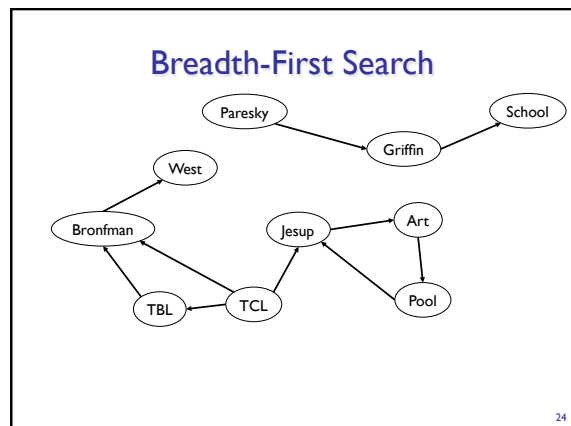


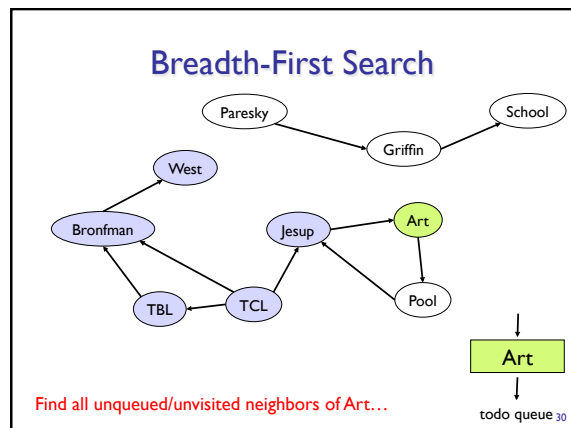
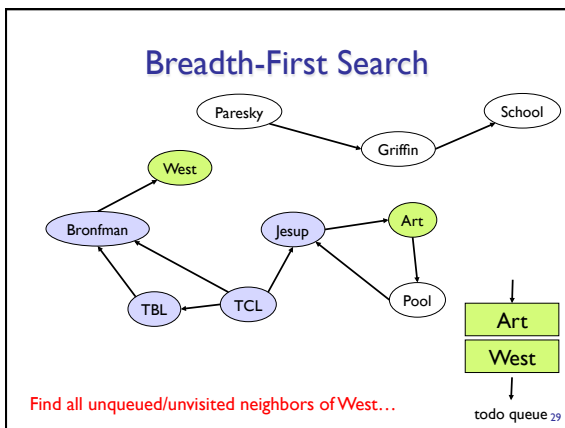
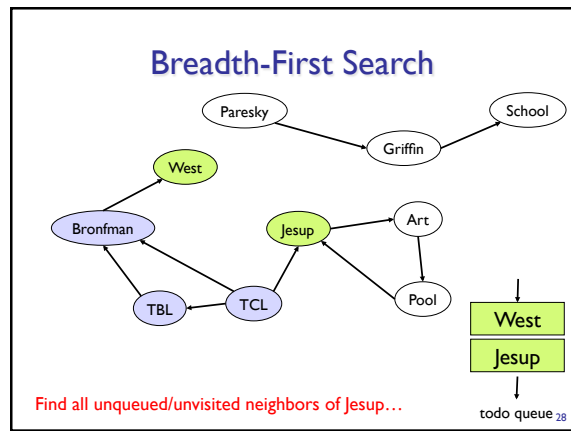
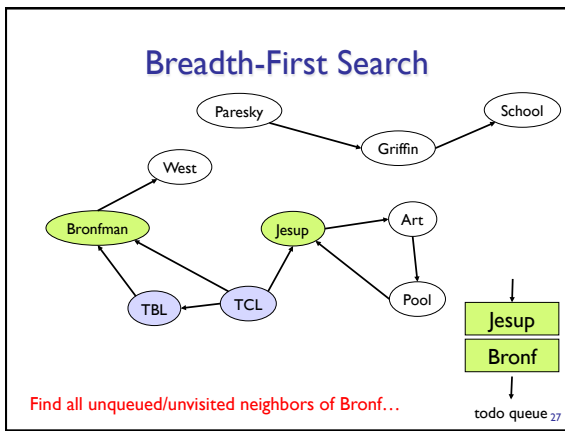
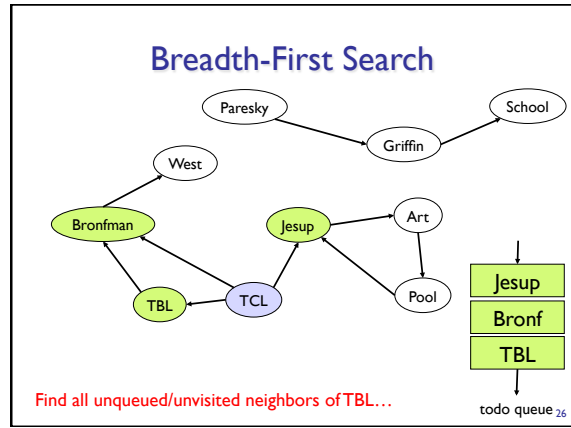
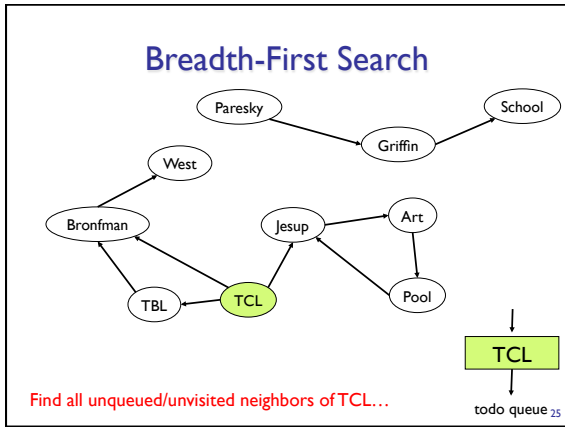
Breadth-First Search

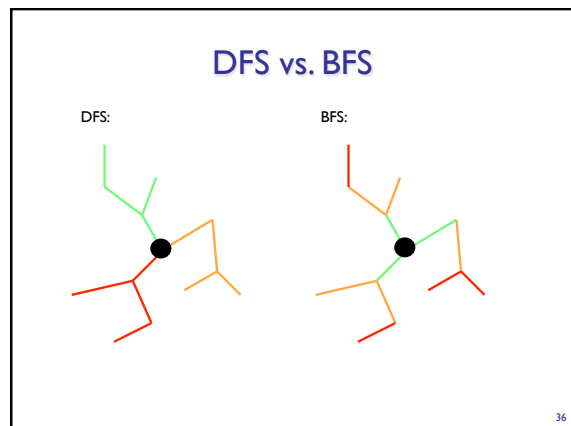
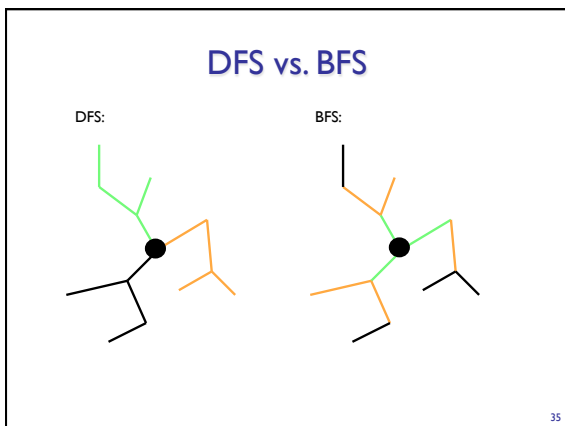
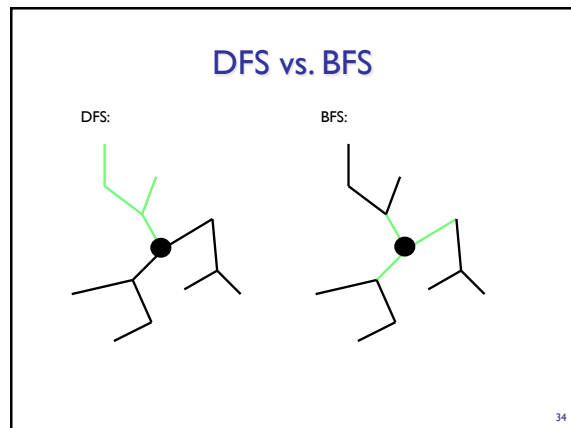
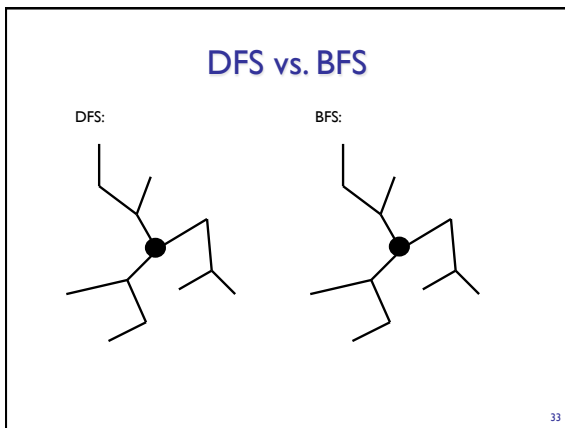
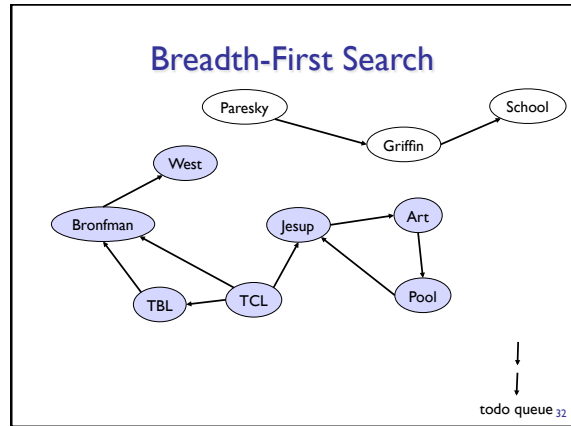
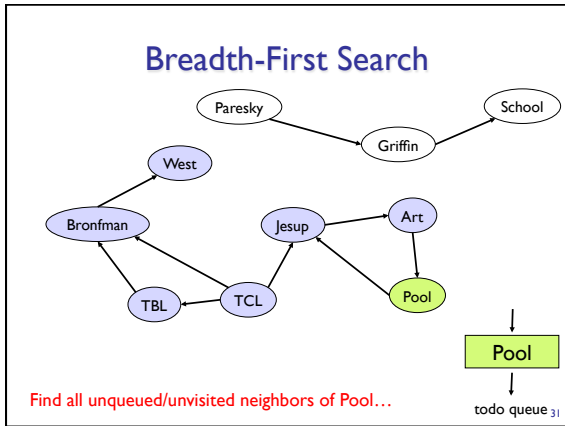
```

void reachableFrom(Graph<V,E> g, V src) {
    Queue todo = new QueueList();
    g.visit(src);
    todo.enqueue(src);
    while (!todo.isEmpty()) {
        V node = todo.dequeue();
        Iterator<V> neighbors = g.neighbors(node);
        while (neighbors.hasNext()) {
            V next = neighbors.next();
            if (!g.visited(next)) {
                g.visit(next);
                todo.enqueue(next);
            }
        }
    }
}
    
```

23







DARWIN!!!!

37