

Lecture 38: Search & Connectivity

Fall 2016

Kim Bruce & Peter Mawhorter

Mentor Applications

- If you want to work as a mentor next semester, please apply *today*
 - Doesn't have to be for this class
- See Piazza for details

Graph Code

- `graph.h` and `graph.c` from this week's assignment

Do they use an adjacency matrix or an adjacency list?

Spanning Trees

- Every tree is a graph...
- ...but every* graph can also be reduced to a tree
- A *spanning tree* is a tree which includes every node of a graph using a subset of that graph's edges
 - *if the graph is not connected, you get *spanning forest*

Spanning Trees

- Can have properties like *minimum-cost*
- Can be constructed by search algorithms

Depth-First Search

- Explore the graph without revisiting nodes
 - *Depth-first* means go until you hit a dead end, then back up to branch out
- Algorithm:
 1. Mark current vertex
 2. Recursively explore all unmarked neighbors
 - (optionally) record where you came from

Depth-First Search

How can we mark a vertex?

How can we record paths?

Use a hash table!

What if we didn't want to use recursion?

Breadth-First Search

What would happen if we replaced the *stack* in DFS with a *queue*?

- Now we explore in order of distance from start
- Algorithm:
 1. Mark start vertex
 2. Add all unmarked neighbors to queue *and mark them*
 3. Repeat step 2 with next from queue until it's empty

BFS Code

(see code examples)

Restarting

- DFS/BFS only explore a single connected component
- To explore entire graph, loop over all vertices and run DFS/BFS again when you find an unmarked one

What is the big-O run time of these algorithms?

$$O(n + m)$$

Testing Connectivity

- For an undirected graph:
 - Run DFS/BFS from any vertex *without restarting* and see if all vertices are marked

Does this work for a directed graph?

Directed Connectivity

- For strong connectivity on a directed graph:
 1. Run D/BFS without restarting from a specific vertex
 2. Run it again from that vertex after reversing all the edges
 - It's strongly connected iff both runs mark all vertices

How could you test weak connectivity?