Announcement

- HW due Friday
- HW due Friday (gradually graded)

CS 314 - Depth First Search
Go until hit a dead end.
What would you do instead?

Backtracking

Why is not a good search strategy.

BFS or not a good search strategy.

Idea: What if your graph is undirected

Depth-First Search (DFS)
Connected. If there are several, then they are:

For all connectedness call DFS(s).

end if

and if

else

Recursively DFS(v).

For each edge (u, v),

mark u

if u is unmarked

Recursively DFS(u).
To traverse a graph, what edge led to the vertex in? What edge led to the vertex in?

end if

if u is unmarked

for each edge (u,v) ∈ E(G)

Recursively DFS(v)

end if

Keep track of the tree.

What do we need to keep track of the tree?

No tree found.

The tree is...

One problem with our code...
Define a function `DFS(s)`

Initialize `S` to be a stack, with `false` for all nodes

While `S` is not empty

Initialize `explored` as `false` for all nodes

while `S` is not empty

Put `s` into `S`

For each edge `(p, u)` in `E`

If `explored [u]` = `false`

Add `(p, u)` to `T`

Else if `explored [u]` = `true`

If `explored [p]` = `false`

Add `p` to `S`; `pop` `S`

End if

End for

End while

End if

End while

`DFS(s)`
For runtime:

Initialization stack operations:

\[ \begin{align*} \sum_{n=1}^{m} & \leq O(m) + O(c) + 0(n) + O(c) + O(n+1) \\ & = O(m^2) + O(c) + O(n+1) \\ & 
\end{align*} \]

How much time do we spend?

Each edge is added exactly \( m - n \) times.

What is put in the stack?
Prop: Let $T$ be a DFS tree $x,y$ be nodes in $T$ s.t. $\exists x,y \in E(T)$.

Then one of $x$ or $y$ is an ancestor of the other (in the tree).

proof is by contradiction (p. 85)
Reurrence for running time:

\[ T(n) = \Theta(n^2) + T(n/2) \]

Quick sort

Examples: Merge sort

But we've already seen this!

Next time: Divide + Conquer (Ch. 5)