Questions about graphs?

- # of paths $s \to t$? ← later in semester
- connected? ← today
- # of edges?
- fastest/shortest/best path $s \to t$? ← next week
- cycles? ← HW
- tree? ← today.

Q for today: is there a path $s \to t$? If so, what is it?

Idea: explore in one "direction" until getting stuck, then backtrack etc.

Mark where we've been to avoid getting caught in cycles.

Mark all vertices UNVISITED { dictionary vertices → bool; array of booleans; set } Θ(1) operations.

[parent ← empty dict]

Depth-first search

```
DFS(G, u):
    Mark u VISITED                          Θ(1)
    for each neighbor v of u:               (assume we can list neighbors of u in Θ(deg(u)))
        if v is UNVISITED:                  Θ(1)
            [parent[v] ← u.]                Θ(1)
            DFS(G, v)
```

Can answer:
- Is graph connected?
- Is s connected to t?

To find a path $s \to t$, use the reversal of t, parent[t], parent[parent[t]], etc.

Time complexity?
- We look @ each vertex at most once. Once visited, we never look @ it again.
- We look @ every edge twice, once from each direction

Hence, overall time complexity is $\Theta(V + E)$

V: # of vertices

E: # of edges.