

Graphs — questions?

- How many cycles? ← *hard!* (are there any?)
 - Longest path? ← *hard*
 - Number of leaves?
 - Is it a tree / is it balanced?
 - Is there a path using every vertex exactly once? ← *hard.*
 - How many vertices?
 - How many connected components? / is it connected?
- ↑ *today.*

Depth-first search (DFS).

Given a graph G and a starting vertex u :

[Mark all vertices unvisited.
 parent ← empty dictionary

— array of booleans,
 hash map, sets, etc.

DFS(G, u):

 Mark u visited.

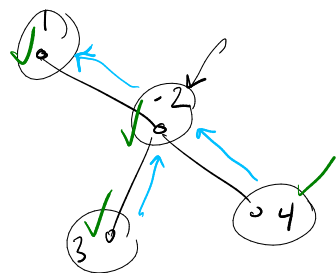
— $\Theta(1)$

 for each neighbor v of u :

 if v is unvisited: — $\Theta(1)$.

 [*parent*[v] ← u — $\Theta(1)$.

 DFS(G, v)



To check if s is connected to t :

• Run DFS(G, s)

• Check if t is visited.

to find a path $s \rightarrow t$ after running DFS(G, s).

look up

look up $\hookrightarrow \text{parent}[t], \text{parent}[\text{parent}[t]], \dots$

① DFS visits each vertex at most once.

- ① DFS visits each vertex at most once
- ② DFS processes each edge at most twice

+ does $O(1)$ operations for each.

So, overall it takes $O(V + E)$ time.

$\uparrow$ $\uparrow$
 # vertices # edges.

Other uses of DFS:

- Articulation points / bridge edges
- Topological sort
- $B(1)$ ancestor queries in a tree

