One issue with BFS.

- Have to search every node of every level before finding $x$.

Ideas? Search as far down as possible. Then go back up.

Depth-First Search ~ pre-order traversal on tree (parent first)

Recursive function
DFS_main \((G, s/\bar{x})\)?

- color all vertices white //not discovered

DFS \((G, s/\bar{x})\)

\[\text{if } v = \bar{x} \text{ return } v\]

- for each white neighbor \(u\) of \(v\):
  - DFS \((G, u/\bar{x})\)
    - color \(v\) black //discovered + all neighbors

[Diagram of a graph with nodes labeled and colored: white, gray, black.]

If Searching for \(x\)?
Run Time?

Still $O(|V| + |E|)$

- every vertex colored white exactly once
- every vertex colored from white to gray exactly once
  - gray, black
- for loop: executed twice for every edge
  - once for $(u,v)$; once for $(v,u)$

Total: $O(3|V| + 2|E|) = O(|V| + |E|)$