Graph traversal and search
traversal
- Depth first search (DFS)
- Breadth first search (BFS)
- shortest path (variation on BFS)

Traversal - visit whole graph
Search - visit graph until you find what you're looking for

Start at 1
do
DFS
depth

DFS
- pick a neighbor
  DFS from that neighbor
repeat (unless I have visited that neighbor)

Order:

1, 0, 5, 2, 3, 4

go as deep as you can
if you hit a dead end or a vertex that has been visited, back up and try a different direction

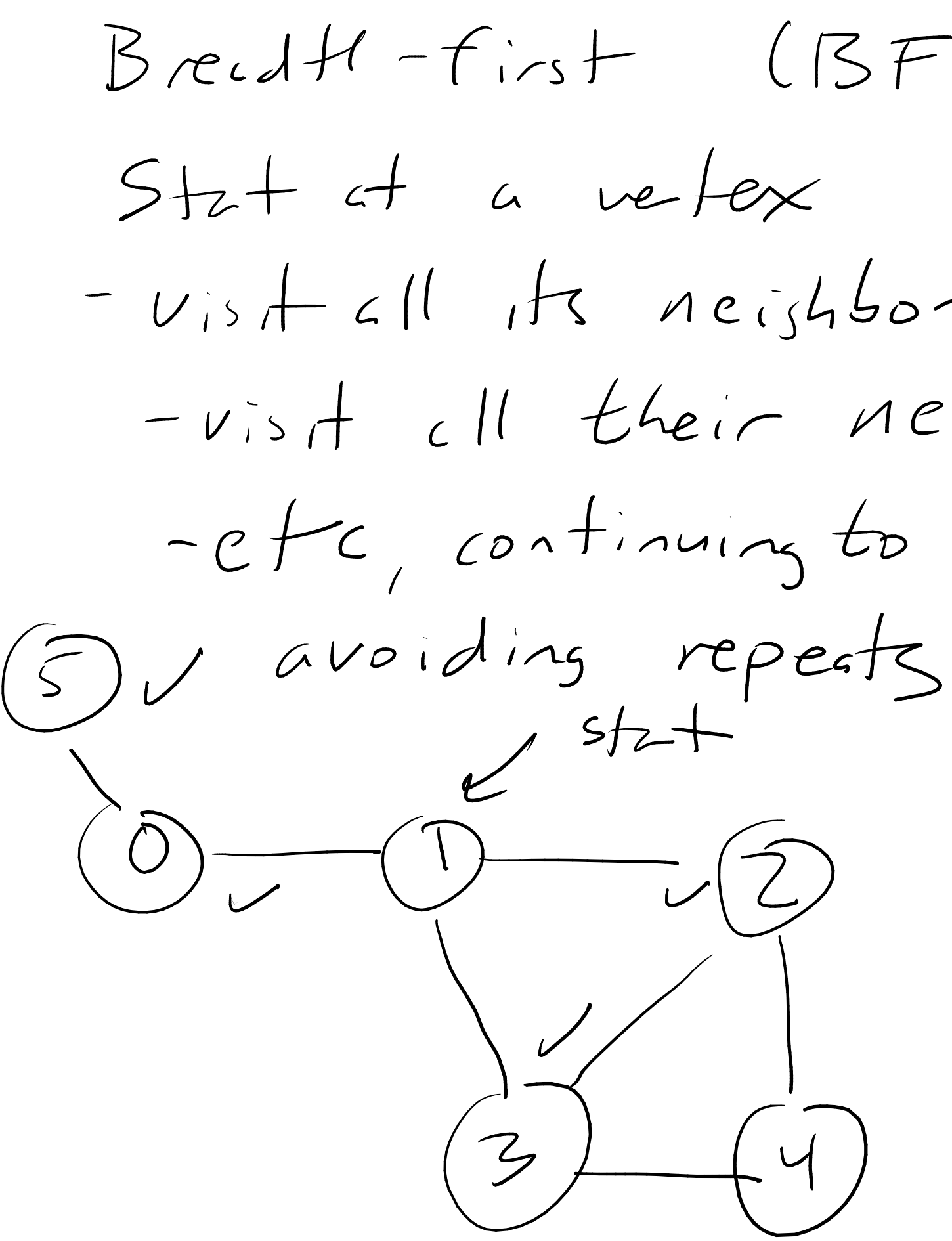
DFS

Given a vertex
loop over all neighbors
visit each neighbor (DFS on that neighbor)
Unless visited previously

Breadth-first (BFS)

Start at a vertex

- visit all its neighbor
  - visit all their neighbors
    - etc, continuing to radiate outwards avoiding repeats



Order:

1, 0, 2, 3, 5, 4 (already saw 3)

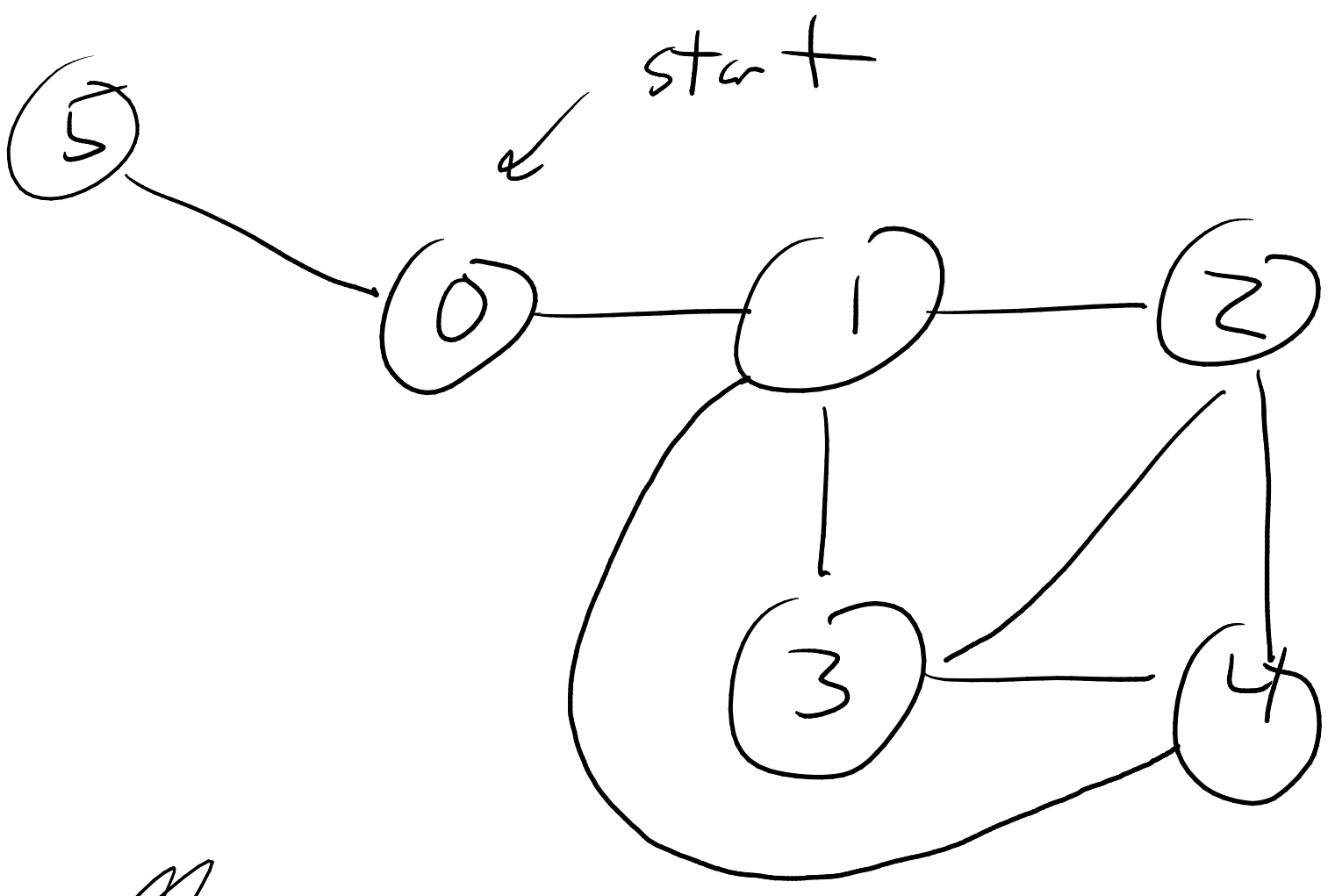
neighbors

BFS typically implemented w/ a queue.

- shortest path is based on BFS

Shortest path is the same as BFS, but keep track of how you got to each node

Find shortest path from 0 to 4



Queue

⊘ → remove and all neighbors

~~5~~ → remove and add all neighbors x

✳ → etc

~~0~~ (no, already)

nghbs of 1 ~~2~~ → no new

3 → no new

4

my destination!

great, what was path?

4 ← 1 ← 0

Visited

0 (via —)

5 (via 0)

1 (via 0)

2 (via 1)

3 (via 1)

4 (via 1)