

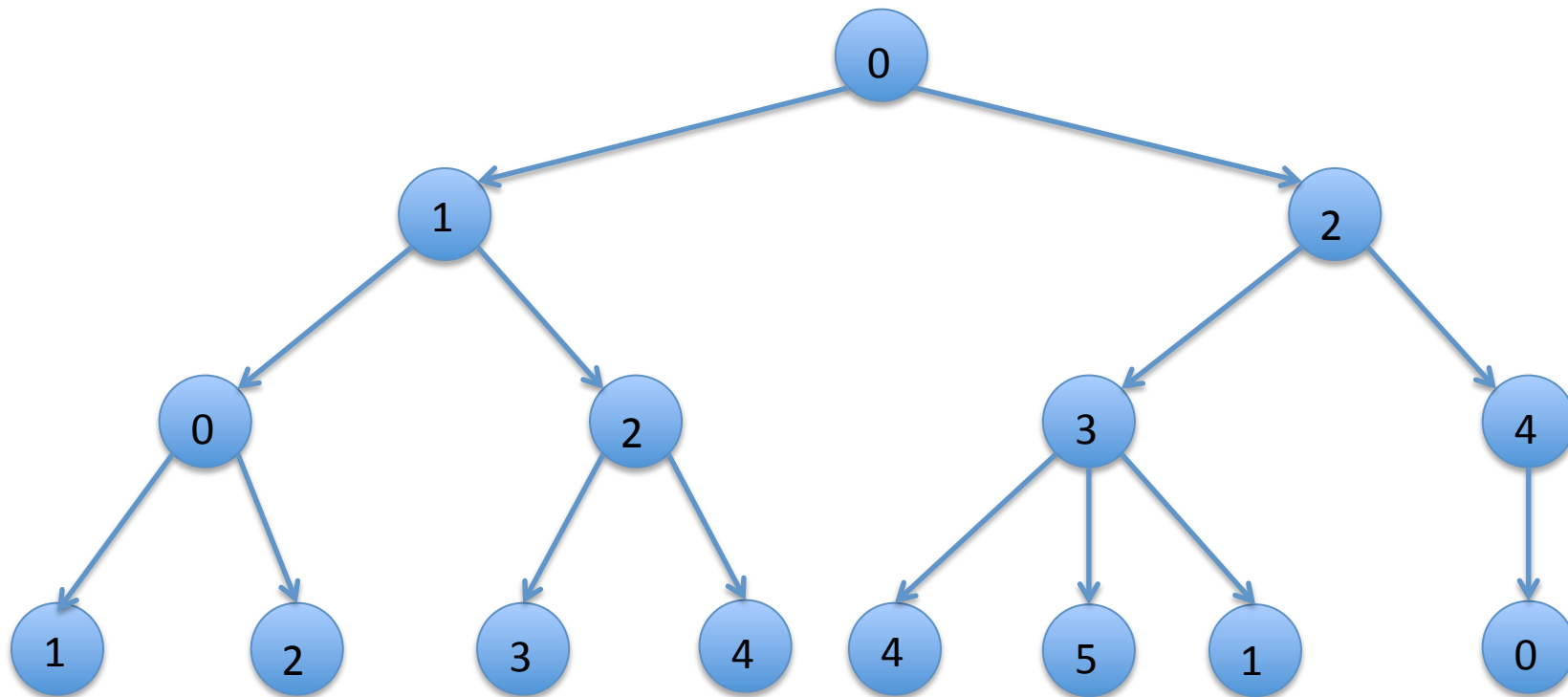
# Search a graph

- Suppose we want to find the shortest path from node 0 to node 5
  - Just in terms of number of steps
  - Or might have weights on edges, and want to minimize total cost
- How might we find this path?

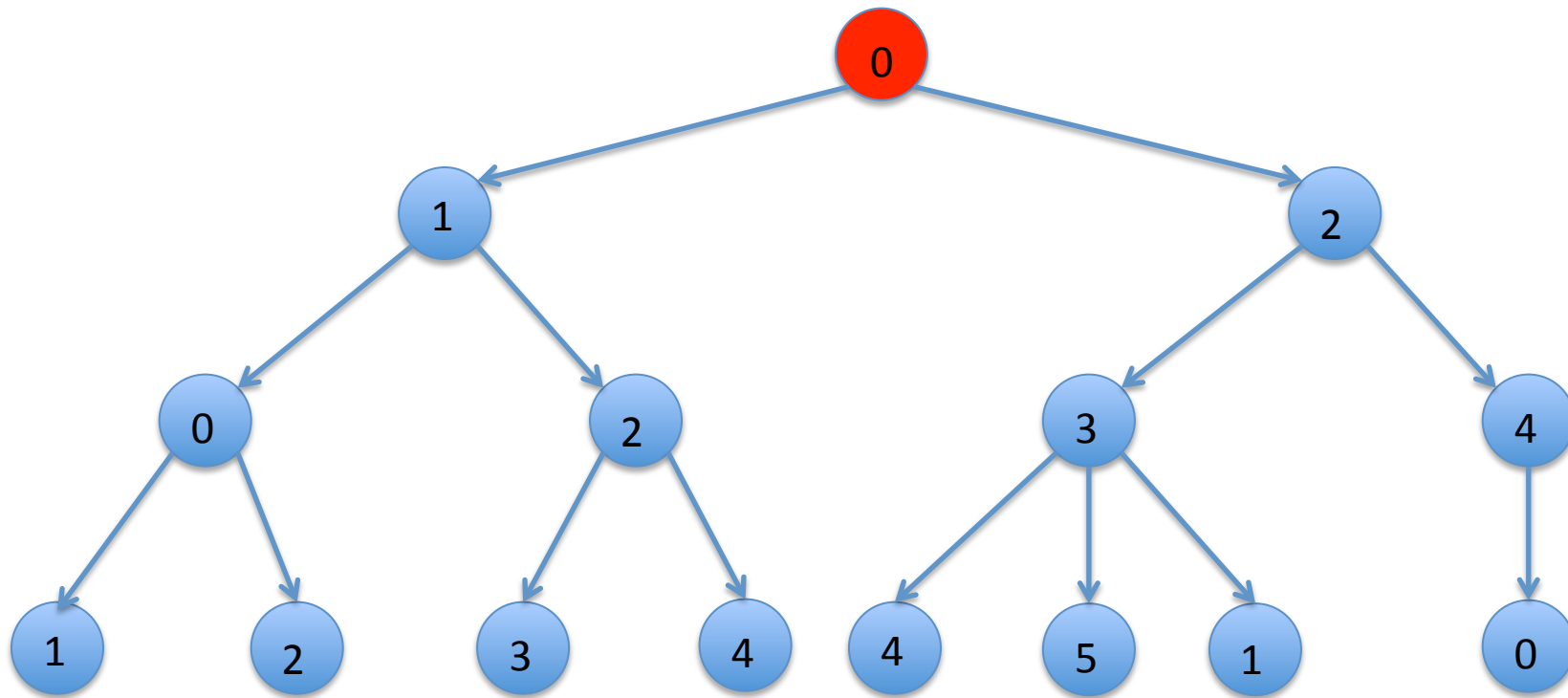
# Depth first search

- Start at “root” node
  - Set of possible paths is just root node
- If not at “goal” node, then
  - Extend current path by adding each “child” of current node to path
  - Add these new paths to potential set of paths, at front of set
  - Select next path and recursively repeat
  - If current node has no “children”, then just go to next option
- Stop when reach “goal” node, or when no more paths to explore

# A tree of solutions to explore

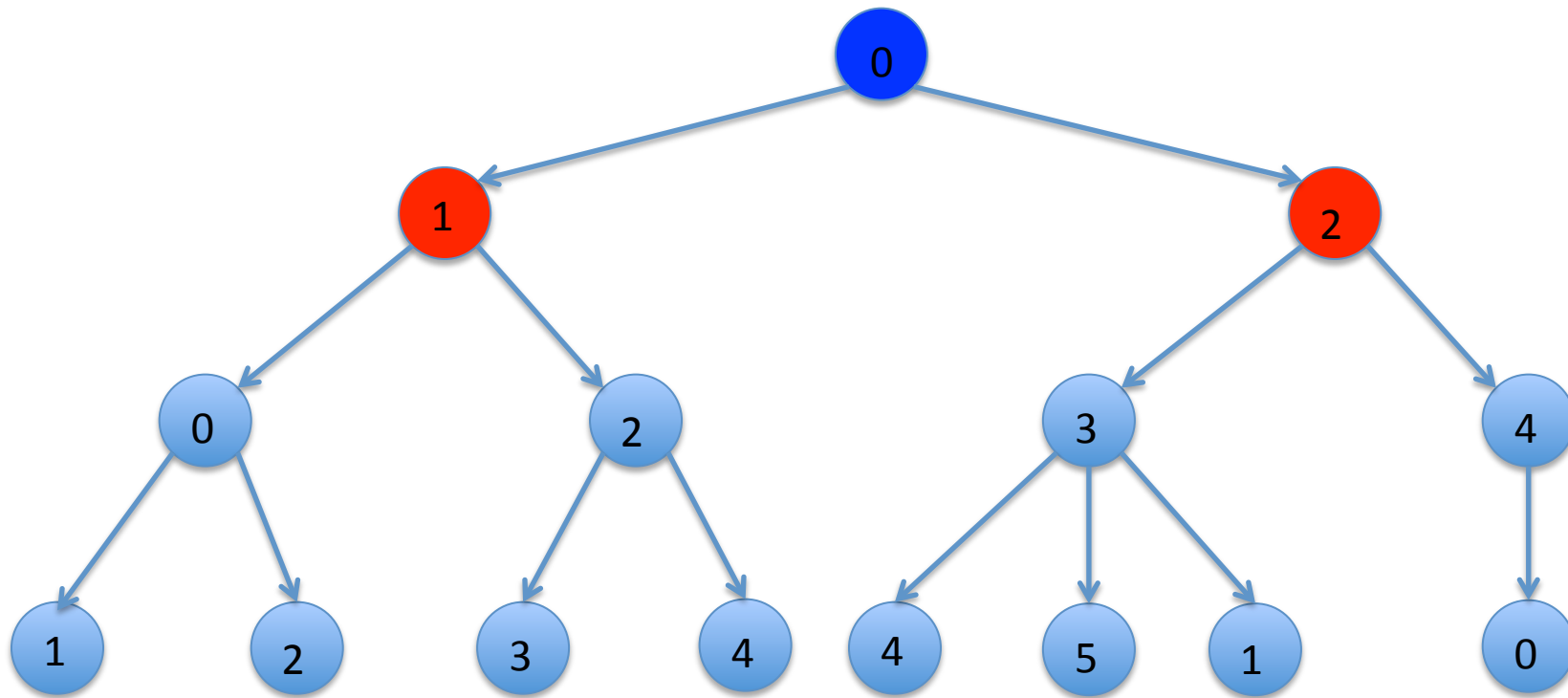


# Simple depth first search



0

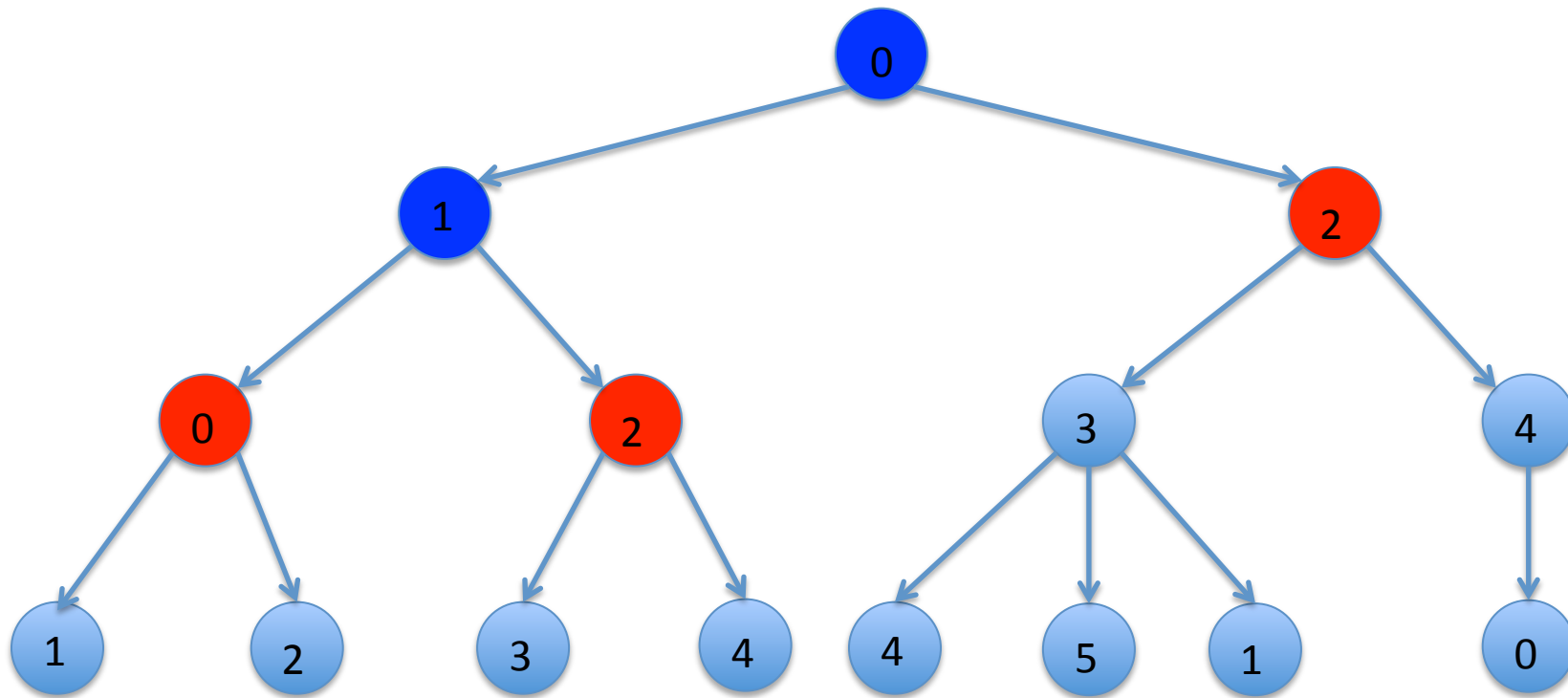
# Simple depth first search



0

01 02

# Simple depth first search

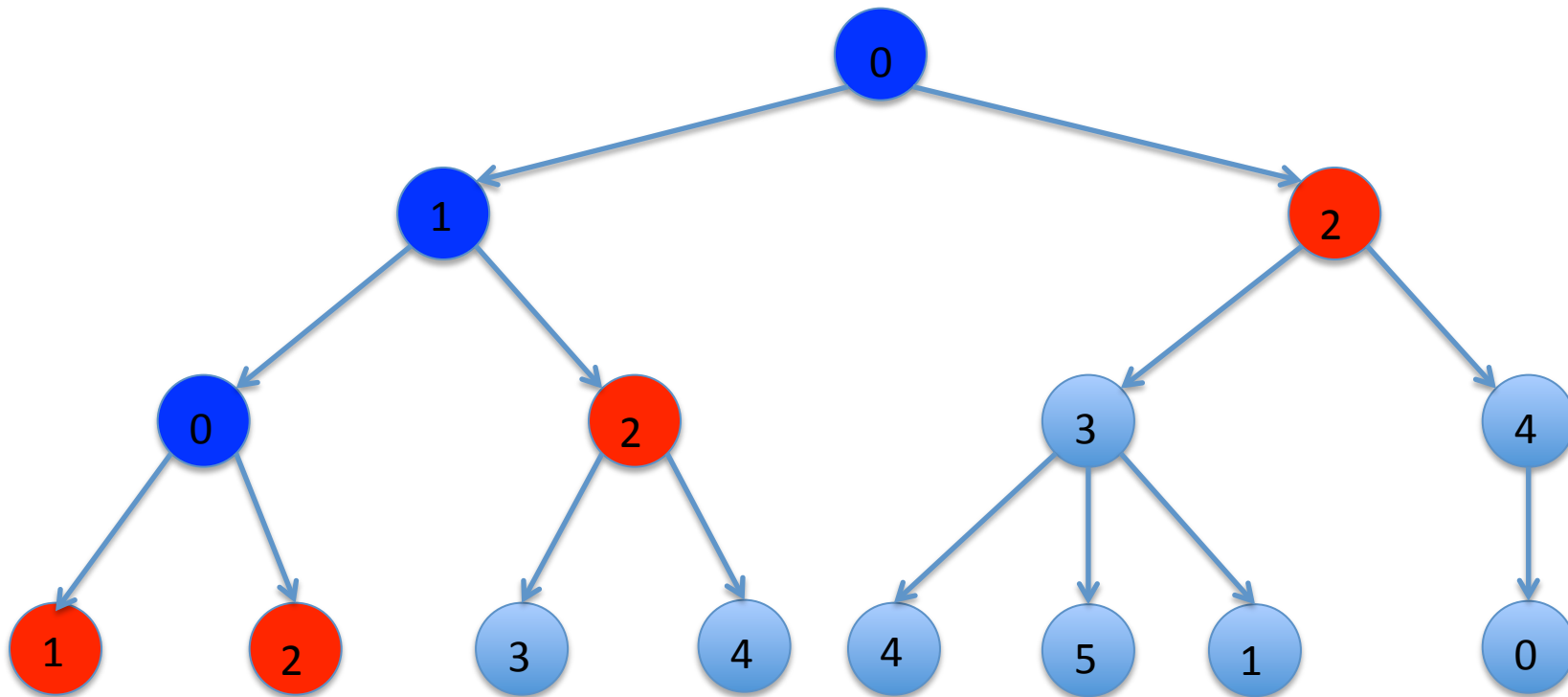


0

01 02

010 012 02

# Simple depth first search



0

01 02

010 012 02

0101 0102 012 02

# We don't want to loop forever!

- If we keep going depth first in this tree, we will just keep cycling between node 0 and node 1
- Avoid visiting a node more than once