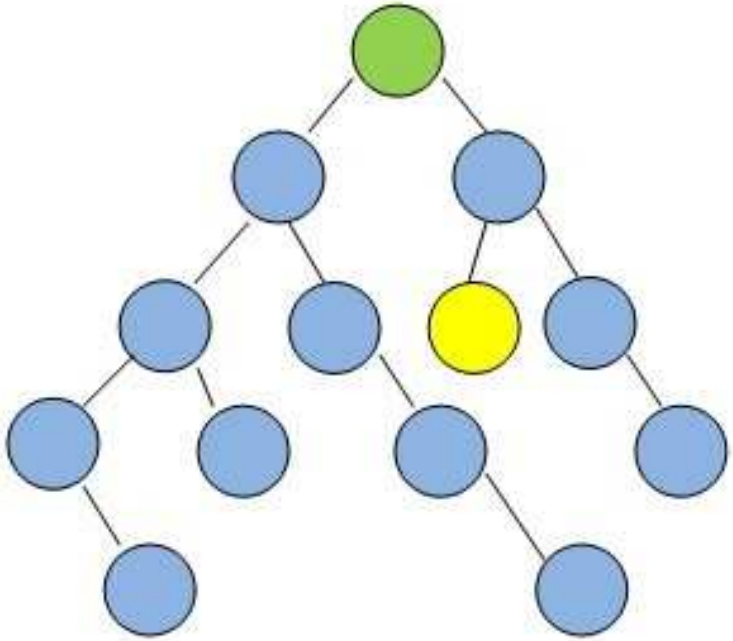
Assignment 12
Sample problems

Graph Search

- In the following graphs, assume that if there is ever a choice amongst multiple nodes, both the BFS and DFS algorithms will choose the left-most node first.



Starting from the green node at the top, which algorithm will visit the least number of nodes before visiting the yellow goal node?

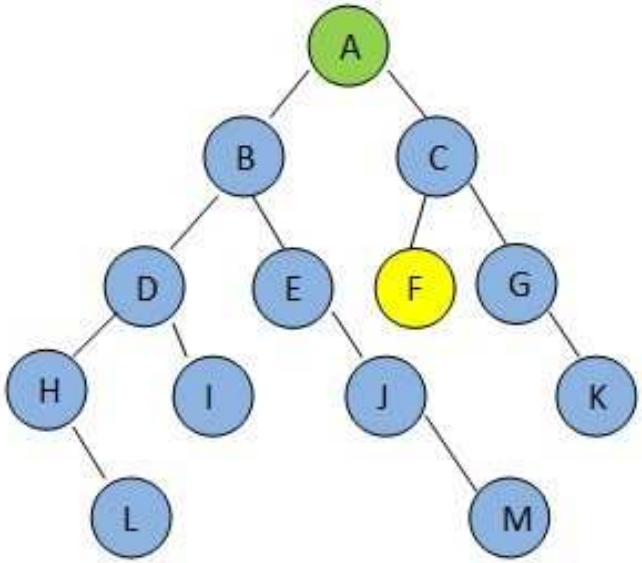
- A: BFS
- B: DFS
- C: Neither BFS nor DFS will ever encounter the goal node in this graph.
- D: BFS and DFS encounter same number of nodes before encounter the goal node

- A: BFS
- B: DFS
- C: Neither BFS nor DFS will ever encounter the goal node in this graph.
- D: BFS and DFS encounter same number of nodes before encounter the goal node

The answer is BFS

How can we get ?

For BFS algorithm, visiting a node's siblings before its children, while in DFS algorithm, visiting a node's children before its siblings

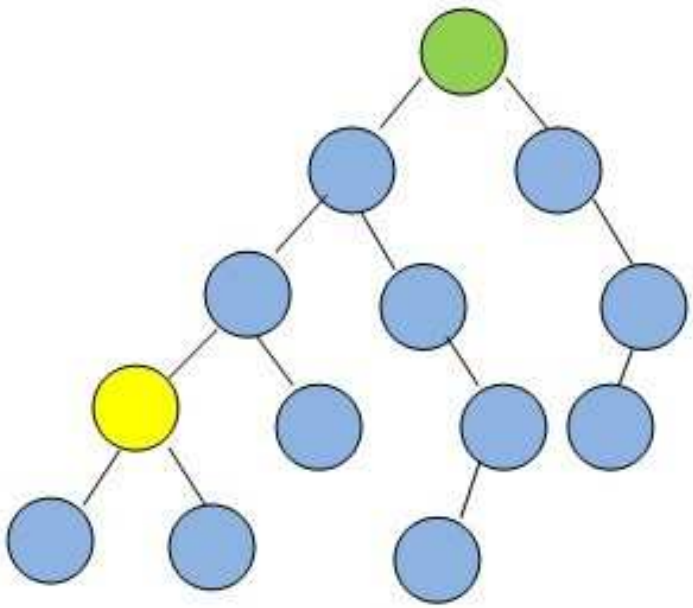


Before countering goal node F:

BFS algorithm encounters nodes: ABCDE

DFS algorithm encounters nodes: ABDHLIEJMC

- In the following graphs, assume that if there is ever a choice amongst multiple nodes, both the BFS and DFS algorithms will choose the left-most node first.



Starting from the green node at the top, which algorithm will visit the least number of nodes before visiting the yellow goal node?

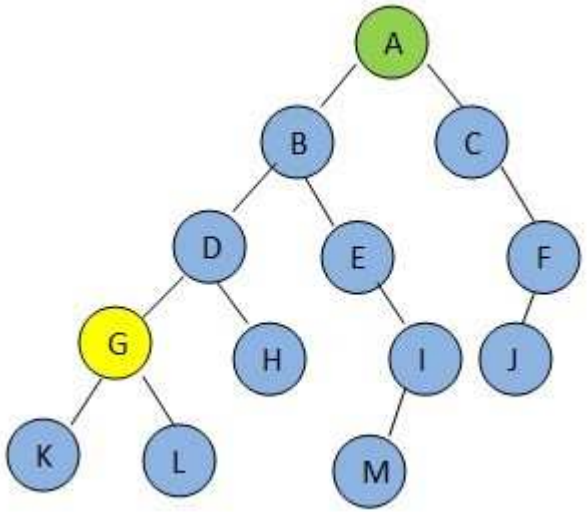
- A: BFS
- B: DFS
- C: Neither BFS nor DFS will ever encounter the goal node in this graph.
- D: BFS and DFS encounter same number of nodes before encounter the goal node

- A: BFS
- B: DFS
- C: Neither BFS nor DFS will ever encounter the goal node in this graph.
- D: BFS and DFS encounter same number of nodes before encounter the goal node

The answer is DFS

How can we get ?

For BFS algorithm, visiting a node's siblings before its children, while in DFS algorithm, visiting a node's children before its siblings

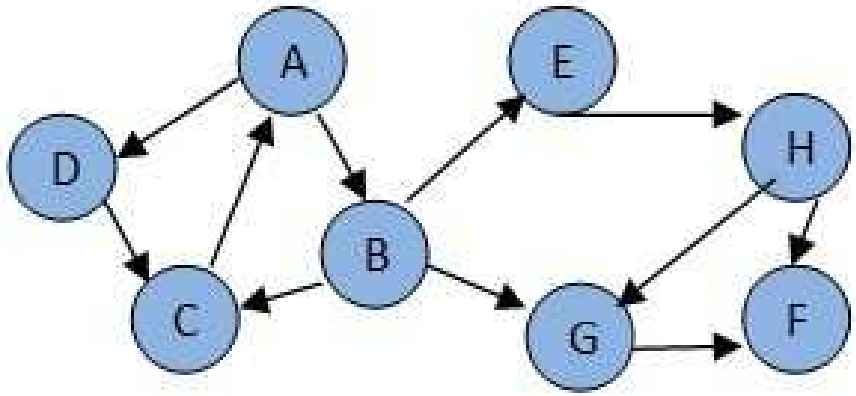


Before countering goal node G:

BFS algorithm encounters nodes: ABCDEF

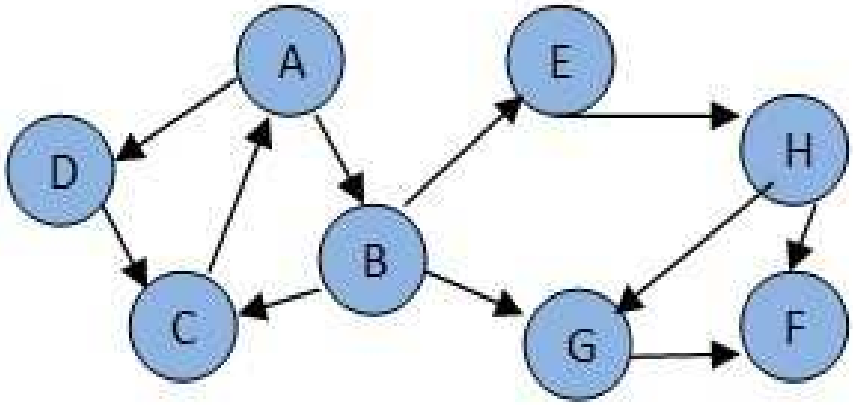
DFS algorithm encounters nodes: ABD

- Consider the following graph. If there is ever a decision between multiple neighbor nodes in the BFS or DFS algorithms, assume we always choose the letter closest to the beginning of the alphabet first.



In what order will the nodes be visited using a Breadth First Search? In what order will the nodes be visited using a Depth First Search?

- Consider the following graph. If there is ever a decision between multiple neighbor nodes in the BFS or DFS algorithms, assume we always choose the letter closest to the beginning of the alphabet first.



In what order will the nodes be visited using a Breadth First Search? The answer is: ABDCEGHF
In what order will the nodes be visited using a Depth First Search? The answer is: ABCEHFGD

How can we get ?

A->B->D

B->C->E->G

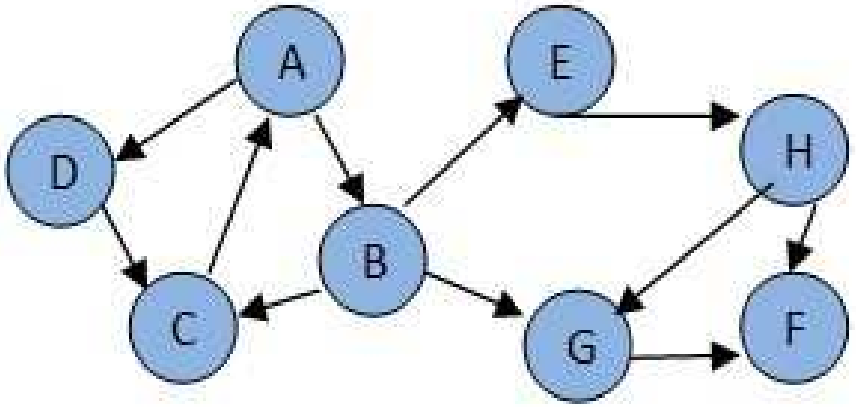
D->C

C->A

E->H

G->F

H->F->G



So for BFS, the answer is ABDCEGHF

for DFS, the answer is ABCEHFGD

Shooting balloon

- Suppose you are playing game of shooting balloon. You expect to shoot n balloons in the board, assuming you are sharpshooter, 100% hit. There are two scenarios, you need find the appropriate Big Oh notation for each scenario. In these problems, one unit of work is shooting one balloon.

- Scenario 1: For every 2 balloons you are able to shoot, one new balloon is inserted in the board. So, if there were 20 balloons, after you shoot the first 2, there are 19 on the board. After you shoot the next 2, there are 18 on the board. How many balloons do you shoot before the board is empty?
- A: $O(1)$
- B: $O(n)$
- C: $O(\lg n)$
- D: $O(n^2)$

- Scenario 1: For every 2 balloons you are able to shoot, one new balloon is inserted in the board. So, if there were 20 balloons, after you shoot the first 2, there are 19 on the board. After you shoot the next 2, there are 18 on the board. How many balloons do you shoot before the board is empty?
- A: O(1)
- B: O(n)
- C: O(lgn)
- D: $O(n^2)$

if One balloon on the board, you shoot 1 balloon in total, if two balloons on the board, you shoot 3 balloons, if three balloons on the board, you shoot 5 balloons in total. So follow it, you shoot 2*n-1 balloons in total.

- Scenario 2: By the time you have shoot the first n balloons, n-1 new balloons have been inserted on the board. After shooting those n-1 balloons, there are n-2 new balloons are inserted on the board. After checking out those n-2 balloons , there are n-3 new balloons on the board. This same pattern continues until on new balloon are inserted on the board. How many total balloons do you shoot before the board is empty?
- A: O(1)
- B: O(n)
- C: O(lgn)
- D: $O(n^2)$

- Scenario 2: By the time you have shoot the first n balloons, n-1 new balloons have been inserted on the board. After shooting those n-1 balloons, there are n-2 new balloons are inserted on the board. After checking out those n-2 balloons , there are n-3 new balloons on the board. This same pattern continues until on new balloon are inserted on the board. How many total balloons do you shoot before the board is empty?
- A: O(1)
- B: O(n)
- C: O(lgn)
- D: $O(n^2)$

$(n+n-1+n-2+n-3+\ldots\ldots+n-(n-1)+n-(n))=(n*n-(n*(n+1)/2))$