

Traversals

Depth-first
search/traversal

Breadth-first
search/traversal

Side by side

Example 1

Example 2

Graph traversals

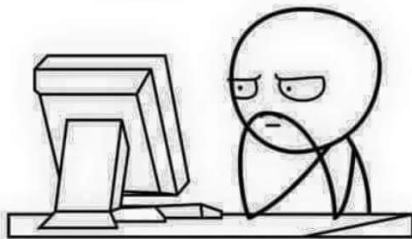
Comp Sci 1575 Data Structures



Traversals

[Depth-first search/traversal](#)[Breadth-first search/traversal](#)[Side by side](#)[Example 1](#)[Example 2](#)

Never let your computer know that you are in a hurry.



Computers can smell fear.
They slow down if they know that
you are running out
of time.

Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

① Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

1 Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

Traversals

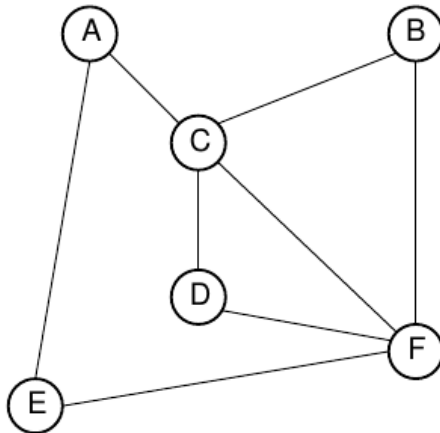
Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2



- How do we define depth-first on this graph with cycles?

Traversals

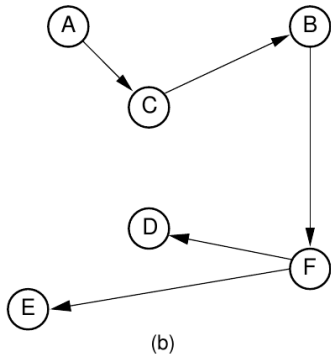
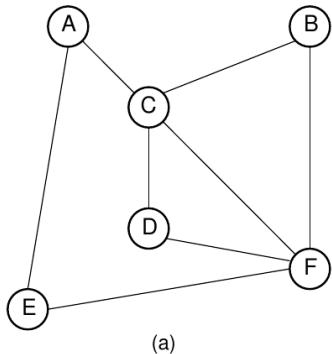
Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2



- a The graph
- b DFS starting at A
 - What is the trace?

Depth-first search: stack frames during recursion

Traversals

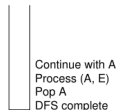
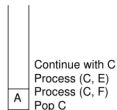
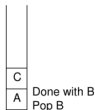
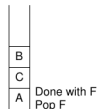
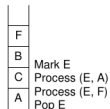
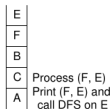
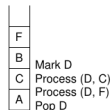
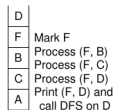
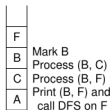
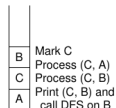
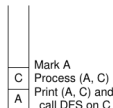
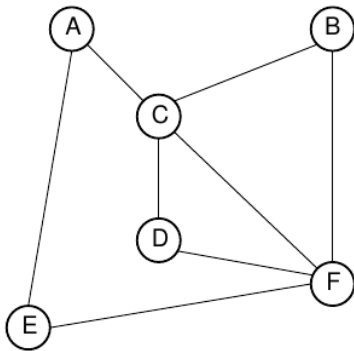
Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2



Traversals

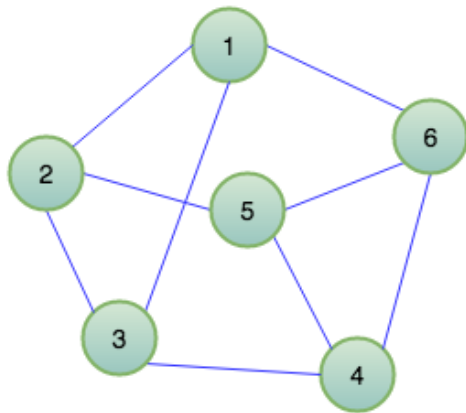
Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2



- How do we define depth-first on this graph with cycles?

Traversals

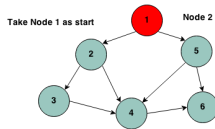
Depth-first search/traversal

Breadth-first search/traversal

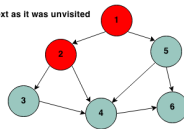
Side by side

Example 1

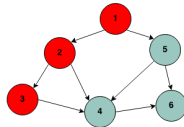
Example 2



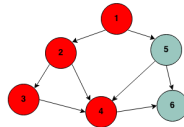
Node 2 is visited next as it was unvisited



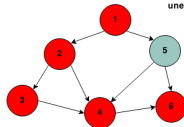
Node 3 is visited next as it was unvisited



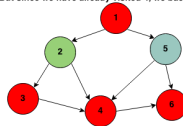
Node 4 is visited next as it was unvisited



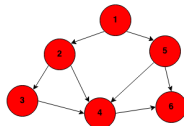
Node 6 is visited next as it was unvisited



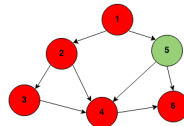
As there is no more nodes which can be traversed from 6, we back track, first node where there is unexplored edge is 2. But since we have already visited 4, we backtrack further



Node 5 from node 1 is visited next as it was unvisited



Again all neighbor nodes from 5 are already visited, hence we back track to node 1



Traversals

Depth-first search/traversal

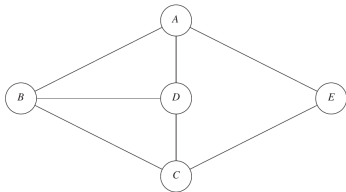
Breadth-first search/traversal

Side by side

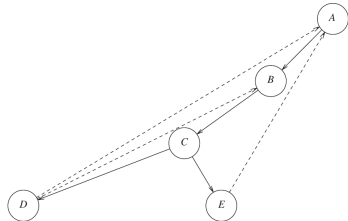
Example 1

Example 2

Left (graph)



Right (DFS)



Traversals

Depth-first
search/traversal

Breadth-first
search/traversal

Side by side

Example 1

Example 2

① Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

Traversals

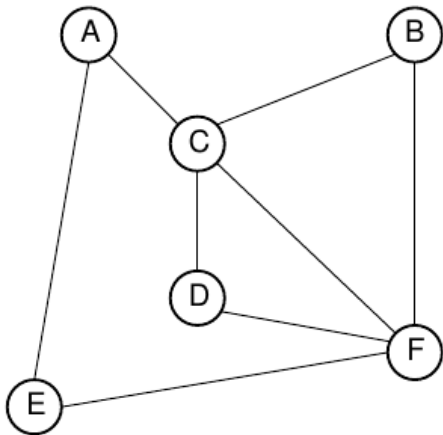
Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2



- How do we define breadth-first on this graph with cycles?

Traversals

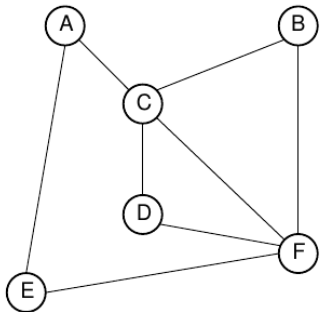
Depth-first
search/traversal

**Breadth-first
search/traversal**

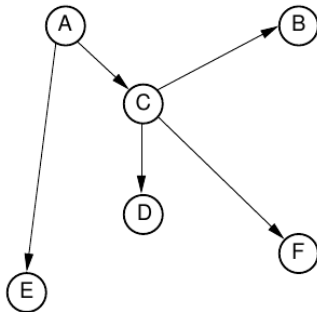
Side by side

Example 1

Example 2



(a)



(b)

- a The graph
- b BFS starting at A
 - What is the order of nodes?

Breadth-first search/traversal: contents of queue

Traversals

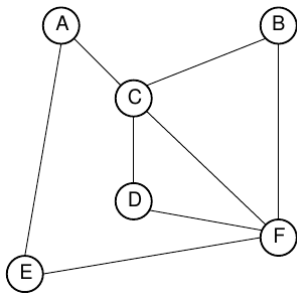
Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

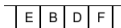
Example 2



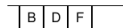
Initial call to BFS on A.
Mark A and put on the queue.



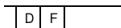
Dequeue A.
Process (A, C).
Mark and enqueue C. Print (A, C)
Process (A, E).
Mark and enqueue E. Print(A, E).



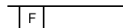
Dequeue C.
Process (C, A). Ignore.
Process (C, B).
Mark and enqueue B. Print (C, B).
Process (C, D).
Mark and enqueue D. Print (C, D).
Process (C, F).
Mark and enqueue F. Print (C, F).



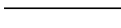
Dequeue E.
Process (E, A). Ignore.
Process (E, F). Ignore.



Dequeue B.
Process (B, C). Ignore.
Process (B, F). Ignore.



Dequeue D.
Process (D, C). Ignore.
Process (D, F). Ignore.



Dequeue F.
Process (F, B). Ignore.
Process (F, C). Ignore.
Process (F, D). Ignore.
BFS is complete.

Traversals

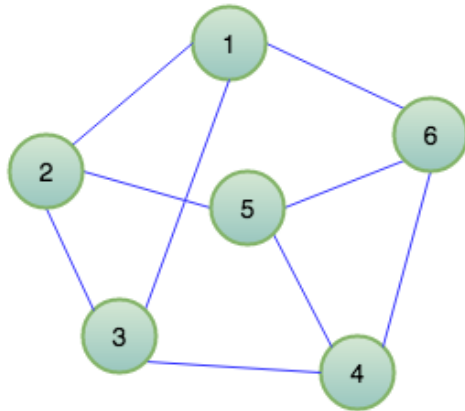
Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2



- How do we define breadth-first on this graph with cycles?

Traversals

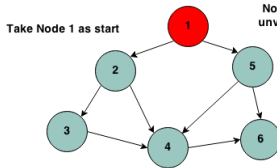
Depth-first
search/traversal

Breadth-first
search/traversal

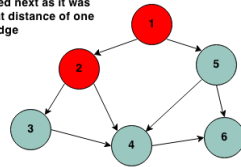
Side by side

Example 1

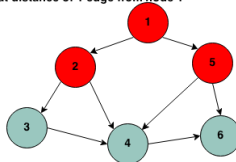
Example 2



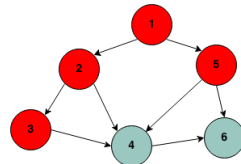
Node 2 is visited next as it was unvisited and at distance of one edge



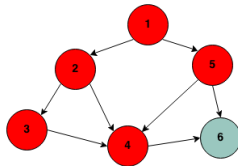
Node 5 is visited next as it was unvisited and at distance of 1 edge from node 1



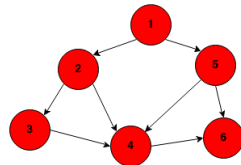
Node 3 is visited next as it was unvisited and at distance of one edge from node 2



Node 4 is visited next as it was unvisited and at distance of one edge from node 2



Node 6 is visited next as it was unvisited and at distance of one edge from node 5



Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

1 Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

Depth-first verses breadth-first search

Traversals

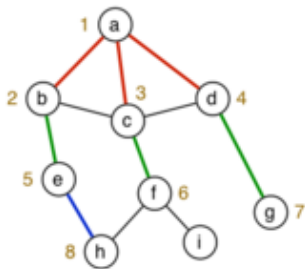
Depth-first search/traversal

Breadth-first search/traversal

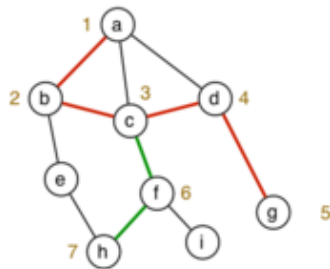
Side by side

Example 1

Example 2

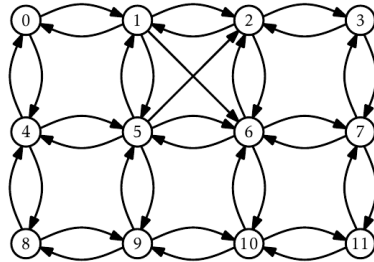


Breadth-first Search



Depth-first Search

Example graph adjacency matrix



	0	1	2	3	4	5	6	7	8	9	10	11
0	0	1	0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	0	1	1	0	0	0	0	0
2	1	0	0	1	0	0	1	0	0	0	0	0
3	0	0	1	0	0	0	0	1	0	0	0	0
4	1	0	0	0	0	1	0	0	1	0	0	0
5	0	1	1	0	1	0	1	0	0	1	0	0
6	0	0	1	0	0	1	0	1	0	0	1	0
7	0	0	0	1	0	0	1	0	0	0	0	1
8	0	0	0	0	1	0	0	0	0	1	0	0
9	0	0	0	0	0	1	0	0	1	0	1	0
10	0	0	0	0	0	0	1	0	0	1	0	1
11	0	0	0	0	0	0	0	1	0	0	1	0

Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

Example graph adjacency list

Traversals

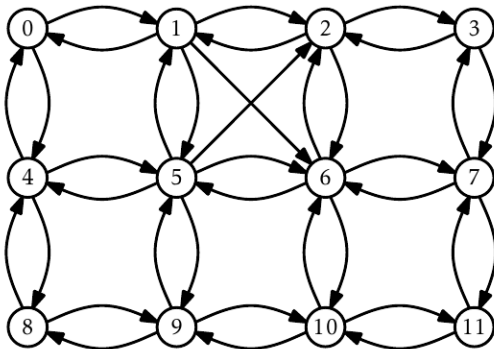
Depth-first
search/traversal

Breadth-first
search/traversal

Side by side

Example 1

Example 2



0	1	2	3	4	5	6	7	8	9	10	11
1	0	1	2	0	1	5	6	4	8	9	10
4	2	3	7	5	2	2	3	9	5	6	7
	6	6		8	6	7	11		10	11	
	5				9	10					
					4						

Traversals

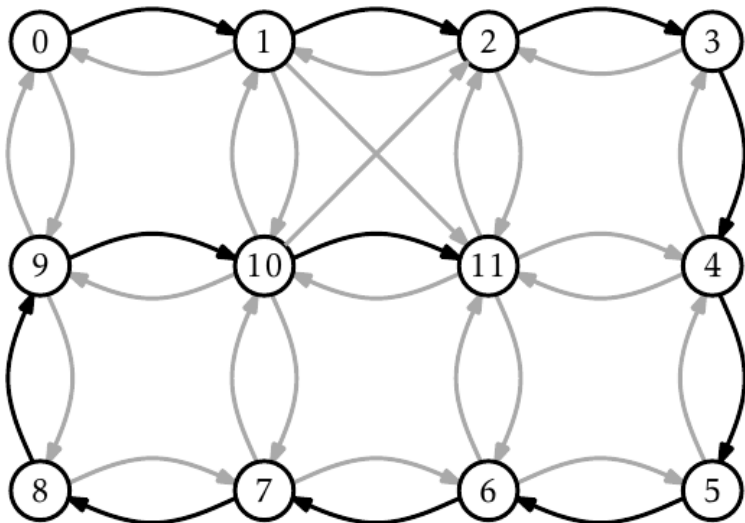
Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2



Traversals

Depth-first search/traversal

Breadth-first search/traversal

Side by side

Example 1

Example 2

