Graph Traversal

BFT and DFT

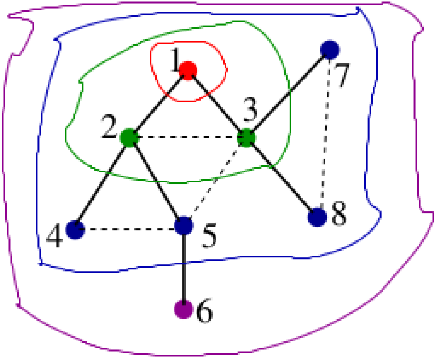
R. Inkulu
http://www.iitg.ac.in/rinkulu/

Outline

1. Breadth-first traversal

2. Depth-first traversal

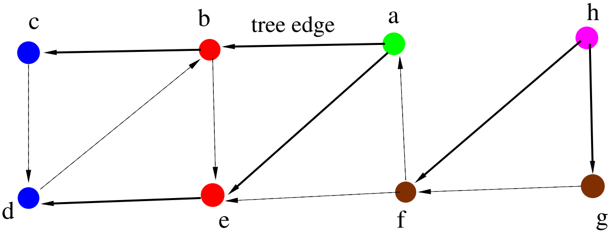
Strategy



$$BFT(1)$$

Initiate an expanding wave from a given node, say 1, and flood the graph with that wave until it grows enough to visit all the nodes that the wave can reach by adding nodes layer by layer, while keeping track of the predecessor vertex of each discovered vertex.

BFT of a directed graph



$BFT(a) \cup BFT(h)$

algo convention: unexplored (white), discovered (grey), finished (black)

colors shown represent layers

takes $O(|V| + |E|)$ time and $O(|V| + |E|)$ space:

- adjacency list representation of $G(V, E)$
- queues (FIFO) for selecting the next node to explore

Invariant of BFT algorithm

At any time, the queue contains vertices corresponding to only two distinct layer ids, in specific, $layerid(tail) \leq layerid(head) + 1$ and the layer ids increase monotonically from head to tail.

- induction on the number of enqueue and dequeue operations

Key properties of BFT[1]

- There exists a path from s to t in G whenever t appears in some layer of $BFT(s)$.

[1] unless specfied otherwise, assume that G is a directed graph whose underlying undirected

Key properties of BFT[1]

- There exists a path from s to t in G whenever t appears in some layer of $BFT(s)$.
- $BFT(s)$ produces an arborescence T_s, known as BFS tree, rooted at s.

[1] unless specfied otherwise, assume that G is a directed graph whose underlying undirected

Key properties of BFT[1]

- There exists a path from s to t in G whenever t appears in some layer of $BFT(s)$.
- $BFT(s)$ produces an arborescence T_s, known as *BFS* tree, rooted at s.
- Let x and y be nodes in T_s belonging to layers L_i and L_j of $BFT(s)$ respectively for $i \leq j$, and let $e(x, y)$ be an edge of an undirected graph G. Then $0 \leq (j - i) \leq 1$.

[1] unless specfied otherwise, assume that G is a directed graph whose underlying undirected

Key properties of BFT (cont)[2]

- In any undirected graph G, each edge joining two nodes in the same layer causes an odd lengthed cycle.

  And, each odd lengthed cycle is due to an edge joining two nodes belonging to the same layer.

  (A graph G is bipartite iff there does not exist an edge joining two nodes in the same layer of $BFT(s)$, for any node s of G.)

[2] unless specfied otherwise, assume that G is a directed graph whose underlying undirected

Key properties of BFT (cont)[2]

- In any undirected graph G, each edge joining two nodes in the same layer causes an odd lengthed cycle.

  And, each odd lengthed cycle is due to an edge joining two nodes belonging to the same layer.

  (A graph G is bipartite iff there does not exist an edge joining two nodes in the same layer of $BFT(s)$, for any node s of G.)

- Suppose $v \in T_s$. Then the simple path from s to v in T_s is a shortest path from s to v in $G(V,E)$.
  - hence, root to leaf paths are as short as possible

[2] unless specfied otherwise, assume that G is a directed graph whose underlying undirected

Key properties of BFT (cont)[2]

- In any undirected graph G, each edge joining two nodes in the same layer causes an odd lengthed cycle.

  And, each odd lengthed cycle is due to an edge joining two nodes belonging to the same layer.

  (A graph G is bipartite iff there does not exist an edge joining two nodes in the same layer of $BFT(s)$, for any node s of G.)

- Suppose $v \in T_s$. Then the simple path from s to v in T_s is a shortest path from s to v in $G(V, E)$.
  * hence, root to leaf paths are as short as possible

- For a directed graph G, a node v belongs to the tree produced by $BFT_G(s)$ if and only if s belongs to the tree produced by $BFT_{G^T}(v)$.

[2] unless specfied otherwise, assume that G is a directed graph whose underlying undirected

Outline

to be continued . . .