Algorithms/Unweighted Graph Algorithms

Top, Chapters: 1, 2, 3, 4, 5, 6, 7, 8, 9, A Please edit and omit unweighted in title

Representation of Graph

Adjacency Matrix

Double plus good.

Adjacency List

HeLLO , difference?!

Comparison

the list might work better with level 1 cache with adjacency objects (which node, visited, inPath, pathWeight, fromWhere).

Depth First Search

Pseudocode

```
dfs(vertex w)
  if w has already been marked visited
    return
  mark w as visited
  for each adjacent vertex v
    dfs(v)
```

Non recursive DFS is more difficult. It requires that each node keep memory of the last child visited, as it descends the current child. One implementation uses a indexed array of iterators, so on visiting a node, the node's number is an index into an array that stores the iterator for the nodes child. Then the first iterator value is pushed onto the job stack. Peek not pop is used to examine the current top of the stack, and pop is only invoked when the iterator for the peeked node is exhausted.

Properties

Classification of Edge

Tree Edge

Backward Edge

Forward Edge

Cross Edge

IT is good techniques from :Yogesh Jakhar

Breadth First Search

Pseudocode

bfs (x):

```
q insert x;
while (q not empty )
   y = remove head  q
   visit y
   mark y
   for each z adjacent y
     q add tail z
```

Example

Correctness

Analysis

Usage

A breadth first search can be used to explore a database schema, in an attempt to turn it into an xml schema. This is done by naming the root table, and then doing a referential breadth first search . The search is both done on the refering and referred ends, so if another table refers to to the current node being searched, than that table has a one-to-many relationship in the xml schema, otherwise it is many-to-one.

Classical Graph Problems

Directed graph cycle detection

In a directed graph, check is *acyclic* by having a second marker on before dfs recursive call and off after, and checking for second marker before original mark check in dfs adjacency traversal. If second marker present, then cycle exists.

Topological Sort of Directed Graph

1. check for cycle as in previous section.
2. dfs the acyclic graph. Reverse postorder by storing on stack instead of queue after dfs calls on adjacent nodes.

The last node on stack must be there from first dfs call, and removing the last node exposes the second node which can only have been reached by last node. Use induction to show topological order.

Strongly Connected Components in Directed Graphs

1. Strong connected components have cycles within and must by acyclic between components (the kernel directed acyclic graph).
2. Difference between dfs reverse postorder of original graph vs the same on reverse graph is that first node is least dependent in latter. Thus all non strong connected nodes will be removed first by dfs on the original graph in the latter's order, and then dfs will remove only strongly connected nodes by marking , one SC component at each iteration over reverse postorder of reverse graph , visiting unmarked nodes only. Each outgoing edge from a SC component being traversed will go to an already marked node due to reverse postorder on reverse graph.

Articulation Vertex

Bridge

Diameter

Top, Chapters: 1, 2, 3, 4, 5, 6, 7, 8, 9, A