Graph Algorithms +6

How can I detect cycles in undirected graph using BFS?

This question previously had details. They are now in a comment.

Answer Follow · 9 Request 2

8 Answers

Jayant Sharma, thedark on codeforces
Answered Aug 2, 2016

You need maintain an array par[i] = parent of node i , now start bfs from node 1 and go on level wise. If a condition occurs when we are exploring neighbours of a node u and it is visited node but is not parent of u , then this is certainly a edge leading cycle.

You can see the code for reference:Ubuntu Pastebin

10.2k Views · View Upvoters

Your feedback is private.

Is this answer still relevant and up to date? Yes No

Upvote · 21 Share

Add a comment... Comment Recommended All

Sravya Vissamsetty, God is REAL, unless explicitly declared INTEGER.
Answered Jul 6, 2016

BFS means a breadth first traversal. It means that you have to visit all the surrounding nodes of a particular node which are unvisited till then and then move on to the next node. continue the process till all the nodes are visited once.

Algorithm:

- **Rule 1** – Visit adjacent unvisited vertex. Mark it visited. Display it. Insert it in a queue.
- **Rule 2** – If no adjacent vertex found, remove the first vertex from queue.
- **Rule 3** – Repeat Rule 1 and Rule 2 until queue is empty.

Related Questions

How can we find all simple cycles in an undirected graph efficiently? How do I find the points that are forming a cycle in a graph?

How can DFS and BFS be used to find out if there are cycles in a graph?

How do I detect a cycle in a directed graph?

I wrote a code to find a cycle inside an undirected graph. It prints if the graph has a cycle or not. How can I make it print the edges of the...

Is it possible to find a cycle in a directed graph using BFS?

In an undirected graph, what is the degree of a vertex with a self-loop?

How do I count the number of cycles in a directed graph?

How do I find if undirected graph is connected or not using DFS?

How can you find if a graph has any cycles using "union-find"?

How can I detect all cyclies in an undirected graph using DFS?

+ Ask New Question

More Related Questions

Question Stats

9 Public Followers

16,017 Views

Last Asked Jul 5, 2016

Edits

If not and no adjacent vertex present, add to code

if(!visited(vertex)) bfs(vertex);

something like this, so that for the next vertex the same procedure continues.

for coding part of bfs go through this link: Breadth First Traversal for a Graph - GeeksforGeeks

If the adjacent node of currect vertex has already been visited, then there is a loop

hope this answers helps you.

6.6k Views · View Upvoters

Shivam Shivam, studied at DAV Public School (2015)
Answered Jun 8, 2017

Well use degree of each node .if degree of the node while adding to queue is greater than the current element whose neighbours are explored their ought to be cycle

While(q is not empty)

Current=deque():

For all neighboring element X of current

{

If deg(X) >deg(current)

Cycle detected

}

1.7k Views

If you need to describe the cycle and not just detect its existence, you need to store the paths as you traverse the graph, which is why DFS might be more convenient.

Just perform BFS while keeping a list of previous nodes at each node visited, or else constructing a tree from the starting node. If I visit a node that has already been marked by BFS, I found a cycle; trace back along the path to describe it.

Remember that either will only work on the connected subgraph of the node you started at, not necessary find cycles on the whole graph.

6k Views · View Upvoters

Upvote · 4 Share

Thomas Cormen, The C in CLRS.
Because the graph is undirected, you also have to check that you're not visiting the no...

Weitao Li
Answered Dec 23, 2017

In bfs you have a visited list, so when you reading neighbors of current node and find there is a neighbor node which was visited before that means you found a loop. Think about this: the reason you reaching current node is because there is a path from root(first node you inserted into queue) to current node, and the reason this neighbor is visited is because there is a path from root to that node, now you found there is an edge between current node and neighbor which means you found a loop. However this does not apply to directed graph.

1.8k Views · View Upvoters

Upvote · 2 Share

Add a comment... Comment Recommended All

Hao Liu, Software Engineer at Backstop Solutions Group (2015-present)
Answered Jul 3, 2017

```
private boolean isCyclicUtil(int v, boolean visited[], Map<Integer, Integer>
mapParentByChild) {

LinkedList<Integer> queue = new LinkedList<Integer>();

visited[v] = true;

queue.add(v);

while (!queue.isEmpty()) {

v = queue.poll();

Iterator<Integer> i = adj[v].listIterator();

while (i.hasNext()) {

int n = i.next();

if (!visited[n]) {
```

```
queue.offer(n);
mapParentByChild.put(n, v);
}
else {
Integer parent = mapParentByChild.get(v);
if (!parent.equals(n)) {
Set<Integer> setParents = new HashSet<Integer>();
setParents.add(n);
Integer child = n;
while (mapParentByChild.get(child) != null) {
setParents.add(mapParentByChild.get(child));
child = mapParentByChild.get(child);
}
while (parent != null) {
if (setParents.contains(parent)) {
return true;
}
parent = mapParentByChild.get(parent);
}
}
}
}
return false;
}
```

2.9k Views

Vance Faber, Problem solver (2012-present)
Answered Jul 3

I have this idea. Use BFS to find a spanning tree keeping track of every edge not on the tree in a list. Each of those edges forms a cycle called a tree cycle and can

364 Views · View Upvoters

Upvote · 1 Share

Add a comment... Comment Recommended All

Top Stories from Your Feed