Algorithm PA3 Report

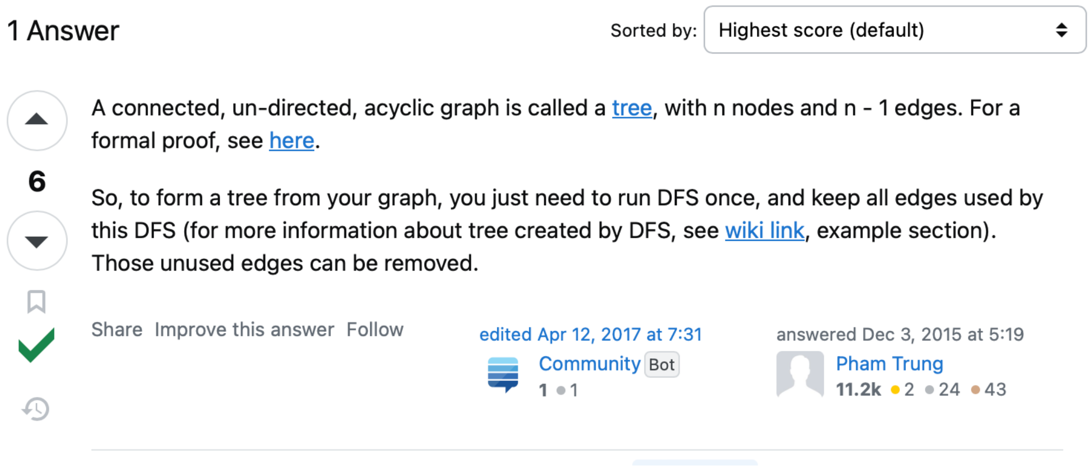
B09502158 詹宜昇

1. Undirected unweighted graph/ undirected weighted graph

   For undirected graph, both weighted and unweighted can be solved by same method. Modify Kruskal's algorithm by looping through the sorted edges in descending order, by doing so we can construct maximum spanning tree instead of minimum spanning tree. To get the minimum removal cost, simply remove those edges don't appear in resulting maximum spanning tree.

   The idea is from:

   https://stackoverflow.com/questions/34055130/removing-edge-that-causes-a-cycle-in-an-undirected-graph


1 Answer
Sorted by: Highest score (default)
6
Share Improve this answer Follow
edited Apr 12, 2017 at 7:31
Community Bot
1 ●1
answered Dec 3, 2015 at 5:19
Pham Trung
11.2k ●2 ●24 ●43

A connected, un-directed, acyclic graph is called a tree, with n nodes and n - 1 edges. For a formal proof, see here.

So, to form a tree from your graph, you just need to run DFS once, and keep all edges used by this DFS (for more information about tree created by DFS, see wiki link, example section). Those unused edges can be removed.

2. Directed weighted graph

   As for directed graph, use DFS to find a cycle in the graph, keep track of the minimum weight edges within a cycle then remove it, by doing so, we can remove an edge in an iteration. Use an infinite loop to keep remove a cycle at every iteration until there exists no cycle in the graph then break.

   The cycle detect algorithm will return True once it detects a cycle, therefore when the algorithm returns False we can break the loop.

   The cycle detect algorithm is from:

   https://www.geeksforgeeks.org/detect-cycle-in-a-graph/

3. Complexity

   Time complexity for undirected graph is the same as Kruskal's algorithm which is O(|E|*log|V|).

   Time complexity for directed graph is O(|cycles|*(|V|+|E|)) which is not polynomial time since it is a NP hard problem.