

CS 9668b Internet Algorithmics

Assignment 3

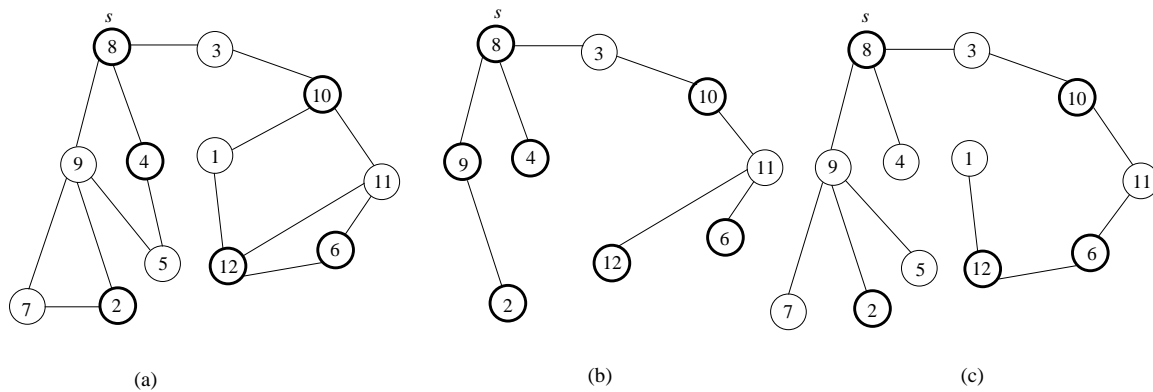
For undergraduate students only

Due date: November 13

Total marks 105 (5 bonus marks).

1. In a synchronous distributed system, the process of sending a message from a root processor s to a subset R of the processors in the system is called *multicasting*.

A multicasting tree is a tree rooted at s that includes all the processors in set R . Some other processors not in R might also belong to the multicasting tree, **but** all the leaves of the multicasting tree must belong to R . For example, for the following network (a) the set R includes all processors with even ID's: $R = \{2, 4, 6, 8, 10, 12\}$. A multicasting tree is shown in (b). The tree in (c) is not a multicasting tree since $7, 5, 4, 1 \notin R$ and processors 7, 5, 4, and 1 are leaves of the tree.

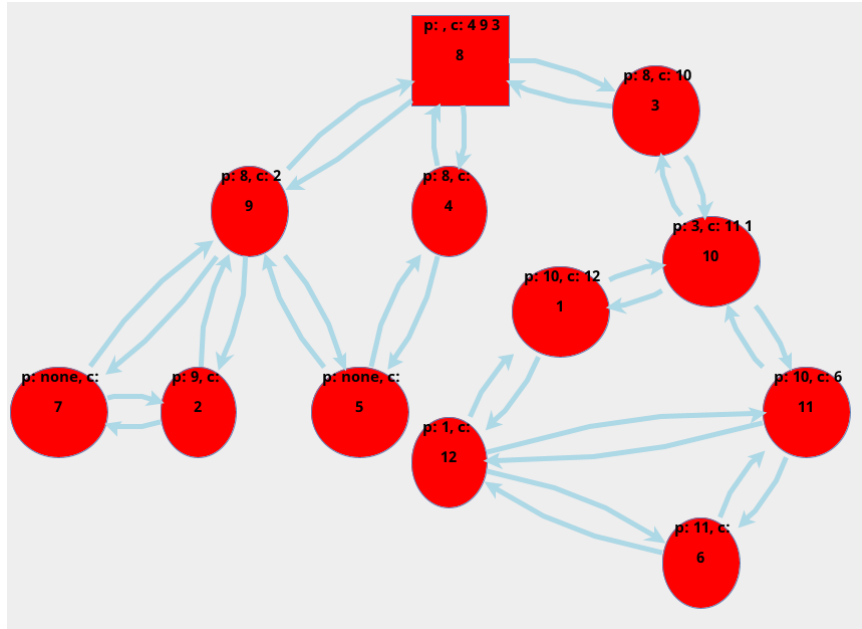


Design and implement in Java a synchronous distributed algorithm that uses flooding to build a multicasting tree rooted at node s for the set R of processors that have even ID's. At the end of the execution of the algorithm each node must know its parent and children in the multicasting tree. Furthermore, each path in this multicasting tree must be a shortest path from s to the other processors in the system.

Assume that the only knowledge that every processor has about the topology of the network is the set of its neighbours. Your algorithm **cannot** assume that a BFS tree is known and it must send at most $O(m)$ messages, where m is the number of edges in the network.

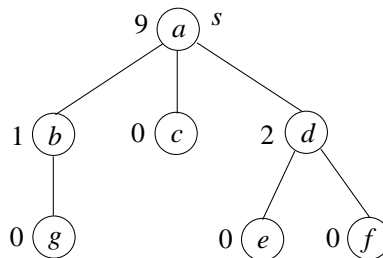
Hints. Modify the BFS tree algorithm discussed in class, but this time all requests for adoption will be acknowledged, whether they are accepted or not. Also this time acknowledgements will not be sent to the potential parent processors right away. A processor might need to wait to hear from all its neighbours before an acknowledgement is sent to its potential parent.

In your Java implementation you will use method `showMessage` to print above each node the id's of its parent and children as shown in the figure below. Use method `isRoot()` to determine which processor is the root processor s .



- (5 marks) Give an informal, high level description of the algorithm in English.
- (25 marks) Submit a Java implementation of your algorithm.
- (5 marks) Prove that your algorithm terminates.
- (10 marks) Prove that your algorithm produces the correct output.
- (5 marks) Compute the time complexity and communication complexity of your algorithm.
- (5 marks) Does your algorithm work on an asynchronous system (without using a synchronizer)? Explain your answer.

2. Given a synchronous distributed system, assume that a BFS tree T rooted at some processor s has already been computed. Design a synchronous algorithm for computing for each node u of the tree the sum of distances from u to each node in the subtree rooted at u . For example, for the following tree, the values that the algorithm must compute for each node are indicated beside the nodes. For leaf nodes, the value is 0. For node b the value is 1 as its subtree has only one node, g , which is at distance 1 from b . The value beside node a is 9 because there are 6 nodes in the tree besides a and the distances from b, c, d to a are 1, from g, e, f to a are 2, so the sum of distances is $1 + 1 + 1 + 2 + 2 + 2 = 9$.



- (5 marks) Give an informal, high level description of the algorithm in English.
- (20 marks) Submit a Java implementation of your algorithm.

In your Java implementation you will use method `showMessage` to print above each node the sum of distances to it from the nodes in its subtree. Use method `isRoot()` to determine which processor is the root processor s .

- (5 marks) Prove that your algorithm terminates.
- (10 marks) Prove that your algorithm produces the correct output.
- (5 marks) Compute the time complexity and communication complexity of your algorithm.
- (5 marks) Does your algorithm work on an asynchronous system (without using a synchronizer)? Explain your answer.