

ASSIGNMENT 3

Subject : Graphs

TAs: Selman Bozkır, Merve Ozdes

Programming Language: Java

Due Date: 24.05.2020 23:59

1 Introduction

Subject of this experiment is usage of graphs. A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph. More formally a Graph can be defined as, A Graph consists of a finite set of vertices(or nodes) and set of Edges which connect a pair of nodes. An example of graph is shown in Figure 1. In this graph, the set of vertices $V = \{0,1,2,3,4\}$ and the set of edges $E = \{01, 12, 23, 34, 04, 14, 13\}$. Graphs are used to solve many real-life problems. Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network. Graphs are also used in social networks like linkedIn, Facebook.

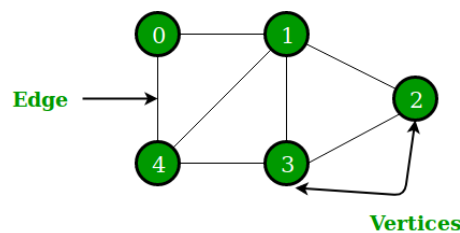


Figure 1: Representation of undirected graph structure

2 PART 1

2.1 Problem Definition

In this part, you will practice about graphs and you will implement election algorithm in wireless environment. Traditional election algorithms are generally based on assumptions that are not realistic in wireless environments. For example, they assume that message passing is reliable and that the topology of the network does not change. These assumptions are false in most wireless environments, especially those for mobile ad-hoc networks. Only few protocols for elections have been developed that work in ad-hoc networks. Vasudevan et al. (2004) propose a solution that can handle failing nodes and partitioning networks. An important property of their solution is that the best leader can be elected rather than just a random as was more or less the case in the previously discussed solutions. Their protocol works as follows.

You should apply election algorithm by using protocol as follows to elect a leader on the graph to be given to you.

- The election is initiated by a source node which is any node in the graph by sending an ELECTION message to its immediate neighbors. (i.e., In Figure 2-a and Figure 2-b, the source node is a and it initiates the election by sending message to its immediate neighbors which are b and j)

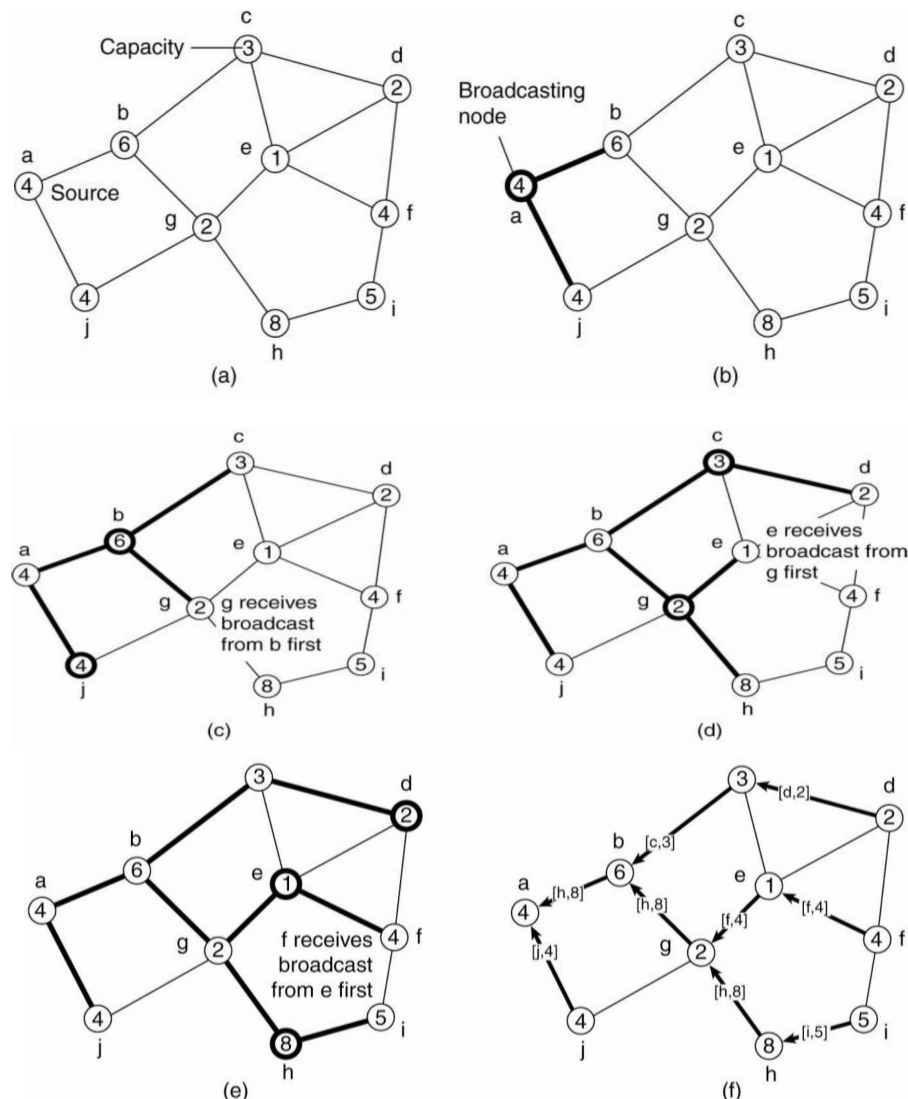


Figure 2: Election algorithm in a wireless network, with node a as the source. (a) Initial network. (b)-(e) The build-tree phase (last broadcast step by nodes f and i not shown). (f) Reporting of best node to source. (This figure from Distributed systems: principles and paradigms (p. 268) Tanenbaum, A. S., & Van Steen, M. (2007))

- If a node receives an ELECTION message for the first time, it identifies the sender as its parent and then sends an ELECTION message to all neighbors except the parent as shown in Figure 2-c.
- When a node receives an ELECTION message from a node other than its parent, it merely acknowledges the receipt.
- If a node has collected acknowledgments from all neighbors except the parent, then it sends an acknowledgment to the parent. This acknowledgment will contain information about the most eligible or the best leader candidate based on resource information of neighbors. Acknowledgments flow back up the tree to the original source.
- Eventually the node that started the election will get all this information and use it to decide on the best leader to be selected – this information can be passed back to all nodes.

This protocol is shown in Figure 2. Each node has been labeled a to j, along with their capacity. You are expected to read the input.txt file and create the graph according to the label and their capacity. In Figure 2, a is source node which initiates an election by broadcasting an ELECTION message to nodes as shown in Figure 2(b). You will be given the starting node which initiates election in the input file. After that step, ELECTION messages are propagated to all nodes, ending with the situation shown in Figure 2(e), where we have omitted the last broadcast by nodes f and i: From there on, each node reports to its parent the node with the largest capacity, as shown in Figure 2(f). For example, when node g receives the acknowledgments from its children e and h, it will notice that h is the best node, propagating [h, 8] to its own parent, node b. In the end, the source will note that h is the best leader and will broadcast this information to all other nodes.

If there are nodes with equal capacities in the graph, there may be more than one best leader. If a node receives the acknowledgments from its child and child's capacity is equal to its own capacity, it reports both its and its child's information to its parent. If a node receives reports from two children of equal capacity, it also reports the information of their two children to the parent node. If this is the case, print all the possible best leader nodes.

2.2 Experiment

In this part, you are supposed to develop a graph and implement a message passing problem. Firstly, you will read input.txt file and create a graph structure according to capacity of each nodes. You are expected to perform the following items.

- Print graph structure with vertices, its capacity and edges.
- Print each broadcast step by nodes
- Print messages sent by nodes to their parents.
- Print the best node/nodes after all messages have been forwarded.

You should print the results of the above items to the output.txt file as shown in Figure 5.

3 PART 2:

In graph theory, if a graph is undirected, connected, and acyclic, it is called a tree. A tree represents the hierarchical structure in a graphical form. Nodes of graphs are called the elements of trees and the edges of the graphs are called branches. A tree with n vertices has $(n-1)$ edges and every edge is a cut-edge. Vertex with a degree of 1 is called a leaf in a tree. The tree is called a balanced tree, if every leaf is "not more than a certain distance" away from the root than any other leaf. and left and right subtrees are balanced. The graph which is shown in Figure 3, is the graph that occurs after all the broadcast steps. This graph is a tree because there is no cycle and this undirected graph is connected.

- In this part, you are supposed to determine the most suitable root of the tree you traveled for broadcast in the first part to provide a balanced tree.
- If there is more than one possible root that satisfies a balanced tree, you should print all of them. Roots that can be selected to be a balanced tree are b and g for the tree shown in Figure 3.

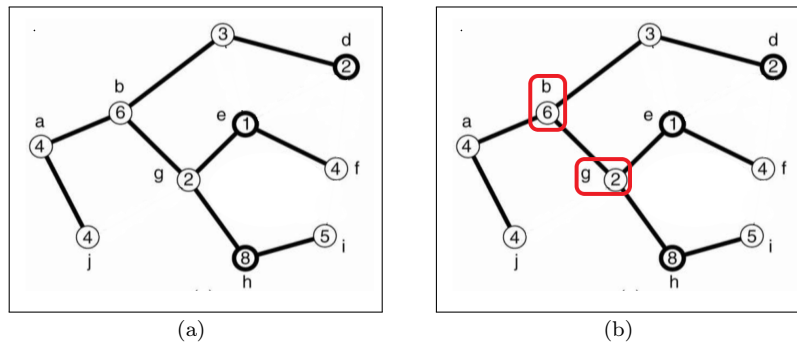


Figure 3: a) The tree formed after all broadcast steps, b) Roots that can be selected to be a balanced tree.

4 Input and Output Format

Your program should read the input file, and construct a graph. The first line of the input file contains each vertex and its capacity. They are separated by a space as shown below. You should create your graph according to this. The second line represents the starting node of the election algorithm. Then the following lines are edges of the graph. Your output file should include the outputs of the above-mentioned items.

```
a 4 b 6 g 2 d 2 e 1 f 4 c 3 h 8 i 5 j 4
a
a b
a j
b g
b c
g e
g h
j g
c d
e f
e c
e d
d f
h i
f i
```

Figure 4: Example Input File

5 Grading and Evaluation

- Your work will be graded over a maximum of 100 points.
- There are two parts you have to do in this experiment and the contribution of each item to your total score will be partial according to the grading policy stated below.
- Your code will be tested with different graph structures. And one output file will be expected as output file that consists the result of each item.
- Your programs are going to be tested on the dev machine.

```

Graph structure:
a(4)-->b j
b(6)-->a g c
c(3)-->b d e
d(2)-->c e f
e(1)-->g f c d
f(4)-->e d i
g(2)-->b e h j
h(8)-->g i
i(5)-->h f
j(4)-->a g
Broadcast steps:
a-->b j
b-->g c
g-->e h
c-->d
e-->f
h-->i
Message passing:
i-->>[i,5]-->>h
f-->>[f,4]-->>e
d-->>[d,2]-->>c
h-->>[h,8]-->>g
e-->>[e,1]-->>g
c-->>[c,3]-->>b
g-->>[g,2]-->>b
j-->>[j,4]-->>a
b-->>[b,6]-->>a
Best node-->h
Possible roots-->g, b

```

Figure 5: Example output file

Reading the file contents and building the graph	10p
Correct broadcast steps	20p
Messages sent by nodes to their parents	20p
Printing best node/nodes after all messages have been forwarded	15p
Printing correct possible roots	20p
Code design and comments	15p

Usage example:

```

>javac Main.java
>java Main input.txt output.txt

```

Notes

- Do not miss the deadline.
- Save all your work until the assignment is graded.
- The assignment must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- Write READABLE SOURCE CODE block
- You can ask your questions via Piazza (<https://piazza.com/hacettepe.edu.tr/spring/bbm204>) and you are supposed to be aware of everything discussed in Piazza.
- You will use online submission system to submit your experiments. <https://submit.cs.hacettepe.edu.tr/> No other submission method (email or etc.) will be accepted. Do not submit any file via e-mail related with this assignment.
- File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system). You must submit your work with the file hierarchy stated below:

```
<Assignment3.zip>/(Required)
  →src/(Required)
    →src/Main.java(Required)
    →*.java(optional)
```

Policy

All work on assignments must be done **individually** unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an **abstract** way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) **will not be tolerated**. In short, turning in someone else's work (from internet), in whole or in part, as your own will be considered **as a violation of academic integrity**. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

- [1] Tanenbaum, A. S., & Van Steen, M. (2007). Distributed systems: principles and paradigms. Prentice-Hall.

- [2] VASUDEVAN, S., KUROSE, J. F., and TOWSLEY, D. F.: "Design and Analysis of a Leader Election Algorithm for Mobile Ad Hoc Networks." Proc. 12th Int'l Conf. on Network Protocols, (Berlin, Germany). Los Alamitos, CA: IEEE Computer Society Press, 2004. pp. 350-360.