



Computing connected components of graphs

Hossein Ebrahimpour-komleh *, Soghra Lazemi

Department of Computer Eng., Faculty of Electrical & Computer Eng., The University of kashan, kashan, Iran

*Corresponding author E-mail: ebrahimpour@kashanu.ac.ir

Copyright © 2014 Hossein Ebrahimpour-komleh, Soghra Lazemi. This is an open access article distributed under the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In this article, we represent an algorithm for finding connected elements in an undirected graph with n vertices based on adjacency matrix.

Keywords: Connected Components, Adjacency Matrix, and Undirected Graph.

1. Introduction

Assume that $G = (V, E)$ is a graph include a V set of vertices, $V = \{1, 2, \dots, n\}$, and is an E set of edges. In all of definitions of graph, it assumes that there is a unique theatrical of vertices and edges. In other words, no vertex (top) and edge more than once will be shown.

A path in a graph $G = (V, E)$ is a sequence of vertices V_1, V_2 that $(V_i, V_{i+1}) \in E$, $1 \leq i \leq k-1$ [1]. The length of such routes is defined as number of edges. A simple path is a path that all of its vertices are individual [1].

Connected components of an undirected graph $G = (V, E)$ is defined as this way: Maximum separated sets $c_1, c_2, \dots, c_k$ that $v = c_1 \cup c_2 \cup \dots \cup c_k$, $u, v \in c_i$ if only be available and V from U can be available, connected components of an undirected graph, are equivalent classes that are under accessibility relation of the vertices [2].

There are some of navigation algorithms for finding connected components [3], [4], [5]. In problems that solution needs finding top target with specified specification, these methods works non-creative.

In this article, the point is achieving set of elements in an undirected graph with n vertices based on adjacency matrix that all of elements are in this set and are connected. An adjacency matrix is preponderated for preparing direct access to information resources on the presence or absence of edges.

2. The proposed method

In proposed method, we use of adjacency matrix for finding connected component directly. The input of algorithm is the adjacency matrix $n \times n$ with n top for an undirected graph and the output of algorithm is connected components.

Step 1: select $i=1$ top (choose a line of graph corresponds with select the vertex of the graph).

Step 2: specify the smallest amount of j that $j=1, 2, \dots, n$, $a_{ij} \neq 0$.

Step 3: assume that a_{ik} is selected in step 2; next selected top is k top. In adjacency matrix, put a_{ik} and a_{ki} equal to zero.

Step 4: are all of elements of k zero now? If yes, connected components of an element are obtained. Otherwise, set $i=k$ and return to step 2.

Step 5: are all of elements of i^{th} row equal to zero? If yes, the connected components of element are obtained. Otherwise, return to step 2.

Step 6: are all elements of adjacency matrix equal to zero? If yes, connected components are obtained.

Proof: After algorithm ends, v_j top is component of L element ($V \in G_L$), if and only if it will be available from existing vertices in L component.

$V_i, V_{i+1}, \dots, V_{i+j}$; $V_i \in G_L$

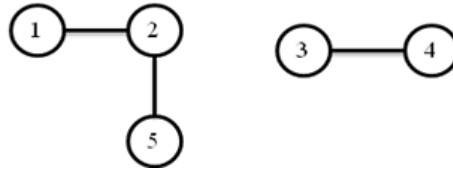
Argument: as it's clear in the algorithm, we know that if one get to the last element in i^{th} row and if $v_{ij}=0$, so, there is a path for i top ($V_i \in G_L$) to j top. Otherwise V_j element belongs to L component ($V_j \in G_L$).

By induction on path length we have:

We assume that there is $V_0, V_1 \dots V_i, V_j \in G_L$ path, that all of corresponding elements in the adjacency matrix are equal to zero. There is induction base for $i=0$ due to above assumption.

Assume that there is a path from V_0 to V_i , that V_i top is formed by edge like (V_{i-1}, V_i) and $V_{i-1, i}=0$. Now, under the above assumption, V_j top is found due to its on end edge that previously had been found and $V_{i,j}=0$. So, there is a path from V_0 to V_j and $V_j \in G_L$. Therefore, it's proved that V_j will belong to G_L element.

Example 1: now, an example of proposed algorithm has been described. Consider the following graph with 5 vertices:



A is a graph adjacency matrix.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Step 1: we choose top number 1 as start point. This top is corresponding with first row of adjacency matrix. The first non-zero component in this row represent next connected top. Because $A(1, 2)=1$, then next top is 2. $A(1, 2)$ and $A(2, 1)$ are equal to zero. Updated adjacency matrix is as follow:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

If all elements of second line are converted to zero, then return to the first row. Otherwise algorithm continues from the second row.

Step 2: in the second row, first non-zero element is $A(2, 5)$. Then, next top is top 5. $A(2, 5)$ and $A(5, 2)$ elements are equal to zero. Updated adjacency matrix is as follow:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

If all elements of 5th row are converted to zero, then we return to the second row, otherwise, algorithm continues from this row.

All elements of 5th row are converted to zero. So the connected components of this graph are {1, 2, and 5}.

If matrix A doesn't have any non-zero element, then the algorithm has ended. Otherwise i have to be updated and algorithm continues from here.

Step 3: in this phase, $i=2$, then we examine the second row. In this row, there isn't any non-zero element. So, i must be updated again and the algorithm continues.

Step 4: in the 3rd row, first non-zero element is $A(3, 4)$, and then next top that should be monitored is forth top. $A(3,4)$ and $A(4,3)$ elements are equal to zero. Updated adjacency matrix is as follow:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 5: all of forth row's elements are equal to zero, and then it continues from third row.

Step 6: all of third row elements are equal to zero, then other elements of connected components are obtained which are {3, 4}.

Step 7: all elements of adjacency matrix are equal to zero, then all the elements of the connected components are obtained and the algorithm ends. ■

The algorithm is written in the pseudo code as follows:

Algorithm CONNECT

Input: the adjacency matrix $n \times n$ with n tops for an undirected graph

Output: connected components of the graph

function CONNECTEDCOMPONENTS(Matrix $A[n,n]$) returns $flag[n]$

$h=1$;

while(there is non zero element in A)

for $j=1:n$ **do**

 Find the smallest j such that $A(i, j) \neq 0$

if j exists **then**

$A(i, j)=0$;

$A(j, i)=0$;

$flag(h)=i$;

$i=j$;

$j=1$;

$h=h+1$;

end

if ($j=n$) **then**

if ($flag \neq null$) **then**

$i=flag(h-1)$;

$flag(h-1)=0$;

$h=h-1$;

else

$i=i+1$;

end

end

end

end

3. Conclusion

In this article we provided a novel algorithm for finding connected components of an undirected graph using adjacency matrix which is more efficient than other algorithms that uses whole graph in their algorithms.

Acknowledgements

Authors are grateful to Research Council of the University of Kashan for providing financial support to undertake this work.

References

- [1] J. Bondy, U. Murty, Graph Theory with Applications, MacMillan, New York, (1976).
- [2] R. Diestel, Graph Theory, 4th Electronic Edition, (2010).
- [3] R. Miller, L. Boxer, Algorithms Sequential and Parallel—a Unified Approach, Second Edition, Charles River Media, (2005).
- [4] D. Jungnickel, Graphs, networks and algorithms, Berlin, Second Auflage, (2005).
- [5] S. Steven, the Algorithm Design Manual, (2011).