Available online at www.elixirpublishers.com (Elixir International Journal)

Discrete Mathematics

Elixir Dis. Math. 37 (2011) 3852-3855

ISSN: 2229-712X

First and second neighborhood for some graphs and its algorithm

Habiba EL-Zohny[1], Randa Salam[2] and Hend EL-Morsy[3]
Department of Mathematics, Faculty of Science, Al-Azahar University, Cairo, Egypt.

ARTICLE INFO

Article history:
Received: 12 June 2011;
Received in revised form:
19 July 2011;
Accepted: 29 July 2011;

Keywords
Neighborhood,
Types of graphs,
Shortest path algorithm.

ABSTRACT

In this paper we compute first and second neighborhood with respect to vertices and edges for some special graphs , and we discussed its algorithm.

Introduction

Difinitions and background:

Definition 1: Degree of vertex:

Let G be an undirected graph or multigraph. For each vertex of G , the degree of v , written deg(v) ,is the number of edges in G that are incident with v .[3]

Definition 2: bipartite graph:

A graph G is bipartite if the set of its vertices can be divided into two disjoint subsets such that each edge has an endvertex in each subset. We denote a bipartite graph by G = (X;□Y;E), where X □and *Y* are the two subsets of vertices (and so XUY □is the set of all vertices) and E is the set of edges.[2]

Definition 3: complete bipartite graph:

A bipartite graph *G* □= (X;□Y;E) is complete if it is simple and the set of its edges is *E*□={ xy│x € X *; y €Y }* that is any pair of a vertex of X and of a vertex of G □is an edge of G . It is denoted by $K_{p,q}$;where p □is the cardinality of X □and q the cardinality of Y.[1]

Definition 4: Cycle graph:

A cycle graph C_n, sometimes simply known as an n-cycle is a graph on $n$nodes containing a single cycle through all nodes. Alternatively, a cycle can be defined as a closed path.[4]

Definition 5: Regular graphs:

A graph G□is said to be regular when the degrees of its vertices are all equal.[2]

Definition 6: Complete graph:

Let v be a set of n vertices, the complete graph on v denoted k_n , is a loop free undirected graph, where for all a,b €v ,a≠b there is an edge {a,b}.[1]

Definition7: weighted graph:

Is a graph for which each edge has an associated real number weight.[5]

Definition 8:

Spanning tree for a graph G is a subgraph of G that contains every vertex of G and is a tree.[5]

Minimal spanning tree for a weighted graph is a spanning tree that has at least possible total weight compared to all other spanning trees for the graphs.[5]

Main Result:

Definition 1:

First neighborhood of vertex v on graph G denoted by N^1(v) or N^1_G(v) is the set of all vertices adjacent to v by one vertex.

Definition 2:

Second neighborhood of vertex v on graph G denoted by N^2_G(v) is the set of all vertices adjacent to v by path of length two.

Definition 3:

N_ neighborhood of vertex v on graph G denoted by N^n_G(v)) is the set of all vertices adjacent to v by path of length n .

Definition 4:

First neighborhood of edge e on graph G denoted by N^1(e) or N^1_G(e) is the set of all edges connect e by one edge.

Definition 5:

Second neighborhood of edge e on graph G denoted by N^2(e) or N^2_G(e) is the set of all edges connect e by path of length two.

Definition 6:

N_ neighborhood of edge e on graph G denoted by N^n(e) or N^n_G(e) is the set of all edges connect e by path of length n .

Neighborhood of vertex v on graph G.

Lemma 1:

First neighborhood of vertex v on graph G equal to the degree of this vertex.

Example 1:

Consider a graph shown in fig(1) ,we can compute first and second neighborhood ,and degree of all vertex as follows:

Vertex	N^1(v)	Deg(v)	N^2(v)
1	N^1(1)={2,5}	2	N^2(1)={4,3}
2	N^1(2)={1,3,5}	3	N^2(2)={4}
3	N^1(3)={2,4}	2	N^2(3)={1,5,6}
4	N^1(4)={3,5,6}	3	N^2(4)={1,2}
5	N^1(5)={1,2,4}	3	N^2(5)={3,6}
6	N^1(6)={4}	1	N^2(6)={3,5}

Tele:
E-mail addresses: hendelmorsy@yahoo.com

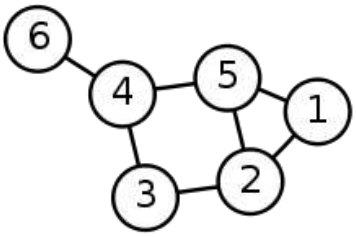


Fig (1)

First and second neighborhood for special graphs:

For complete bipartite graph:

There are many types of complete bipartite graphs $K_{m,n}$.

Cas(1): when m=n

Example 2:

We can compute first and second neighborhood for complete bipartite graph shown in fig(2) as follows:

Vertex	$N^1(v)$	Deg(v)	$N^2(v)$
V1	{v2,v4,v6}	3	{v5,v3}
V2	{v1,v3,v5}	3	{v4,v6}
V3	{v2,v4,v6}	3	{v1,v5}
V4	{v1,v3,v5}	3	{v2,v6}
V5	{v2,v4,v6}	3	{v1,v3}
V6	{v1,v3,v5}	3	{v2,v4}

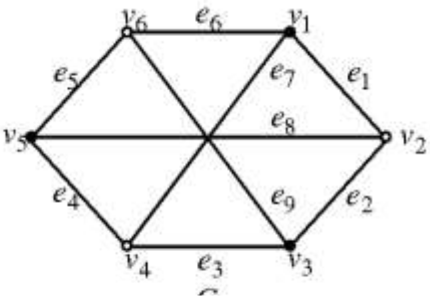


Fig (2)

Example 3:

For fig(3) ,$K_{4,4}$ we have:

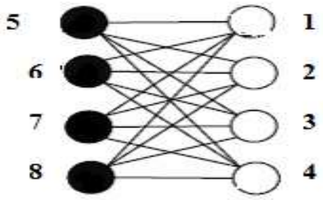


Fig (3)

Vertex	$N^1(v)$	deg(v)	$N^2(v)$
1	{5,6,7,8}	4	{2,3,4}
2	{5,6,7,8}	4	{1,3,4}
3	{5,6,7,8}	4	{1,2,4}
4	{5,6,7,8}	4	{1,2,3}
5	{5,6,7,8}	4	{6,7,8}
6	{5,6,7,8}	4	{5,7,8}
7	{5,6,7,8}	4	{5,6,8}
8	{5,6,7,8}	4	{5,6,7}

Cas(2): when m≠n

Example 4:

Consider bipartite graph $K_{3,2}$ with m=3, n=2 as shown in fig(4).

Vertex	$N^1(v)$	Deg(v)	$N^2(v)$
1	{4,5}	2	{2,3}
2	{4,5}	2	{1,3}
3	{4,5}	2	{1,2}
4	{1,2,3}	3	{5}
5	{1,2,3}	3	{4}

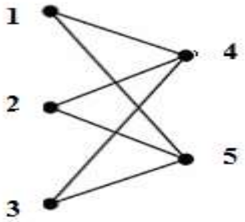


Fig (4)

Theorem 1:

For complete bipartite graph $K_{m,n}$,

for m=n

i. first neighborhood for every vertex on set m is equal to all vertices on a set n ,and converse is true.

ii.second neighborhood for every vertex is equal to (m-1) or (n-1).

For m≠n

i. first neighborhood for every vertex on set m is equal to all vertices on a set n . and the converse is true.

ii.second neighborhood for every vertex on set m is equal to (m-1),and for every vertex on a set n is equal to (n-1).

Proof:

The proof comes directly from the above discussion.

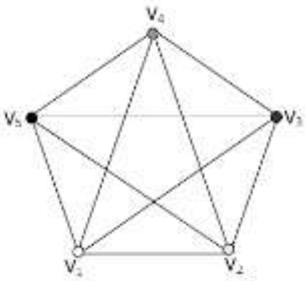
For regular graph:

We can compute first and second neighborhood for regular graph on the same way.

Example 5:

Consider 4-regular graph as shown in fig (5),we have:

Vertex	$N^1(v)$	Deg(v)	$N^2(v)$
V1	{v2,v3,v4,v5}	4	{v2,v3,v4,v5}
V2	{v1,v3,v4,v5}	4	{v1,v3,v4,v5}
V3	{v1,v2,v4,v5}	4	{v1,v2,v4,v5}
V4	{v1,v2,v3,v5}	4	{v1,v2,v3,v5}
V5	{v1,v2,v3,v4}	4	{v1,v2,v3,v4}



Fig(5)

Example 6:

For 3-regular graph shown in fig(6) we have:

Vertex	$N^1(v)$	Deg(v)	$N^2(v)$
V1	{v2,v3,v4}	3	{v2,v3,v4}
V2	{v1,v3,v4}	3	{v1,v3,v4}
V3	{v1,v2,v4}	3	{v1,v2,v4}
V4	{v1,v2,v3}	3	{v1,v2,v3}

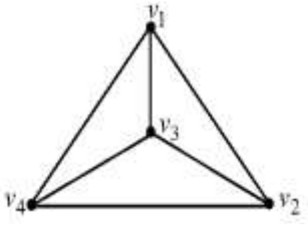


Fig (6)

Theorem 2:

For regular graph K-regular. First neighborhood for each vertex equal to second neighborhood, $N^1(v)=N^2(v)$.

For complete graph:

All results discussed on regular graph is the same on complete graph.

Example 6:

For K_4 (regular graph with 4 vertices) shown in fig(7),we have:

Vertex	$N^1(v)$	Deg(v)	$N^2(v)$
1	{2,3,4}	3	{2,3,4}
2	{1,3,4}	3	{1,3,4}
3	{1,2,4}	3	{1,2,4}
4	{1,2,3}	3	{1,2,3}

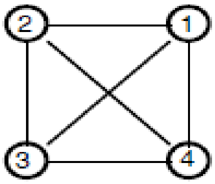


Fig (7)

For cyclic graph:

We have two cases on cycle graph, n-cycle.

Cas (1) for n=3 ,(3-cycle)

Example 7:

Consider cycle graph with 3 vertices(3-cycle) fig(8), we can compute first and second neighborhood as follows:

vertex	$N^1(v)$	Deg(v)	$N^2(v)$
1	{2,3}	2	non
2	{1,3}	2	non
3	{1,2}	2	non

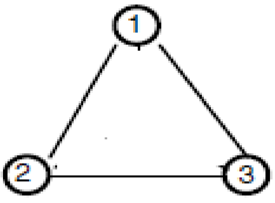


Fig (8)

Cas (2): for n>3

Example 8:

Consider cycle graph (4-cycle) with 4 vertices shown in fig (9),

vertex	$N^1(v)$	Deg(v)	$N^2(V)$
1	{2,4}	2	{3}
2	{3,1}	2	{4}
3	{2,4}	2	{1}
4	{1,3}	2	{2}

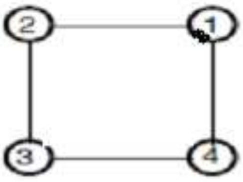


Fig (9)

Example 9:

For 5-cycle graph C_5 with 5 vertices shown in fig (10),

Vertex	$N^1(v)$	Deg(v)	$N^2(v)$
V1	{v3,v5}	2	{v3,v4}
V2	{v1,v3}	2	{v1,v5}
V3	{v2,v4}	2	{v1,v5}
V4	{v3,v5}	2	{v1,v2}
V5	{v1,v4}	2	{v2,v3}

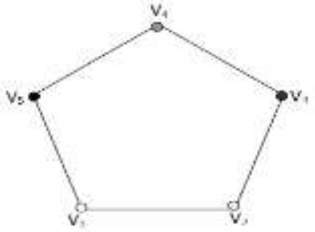


Fig (10)

Theorem 2:

For cycle graph n-cycle we have two cases,

Cas (1): for n=3

i. First neighborhood for each vertex equal to (n-1).

ii. Second neighborhood doesn't exist.

Cas (2): for n>3

i. First neighborhood for each vertex equal to 2.

ii. Second neighborhood for each vertex equal to (n-3).

Proof:

The proof comes directly from the above discussion.

Neighborhood for edge e on graph G:

Example 10:

For bipartite graph shown in fig(2),

Edge	$N^1(e)$	$N^2(e)$
e 1	{e2,e6,e7,e8}	{e3,e4,e5}
e2	{e1,e3,e8,e9}	{e4,e5,e6}
e3	{e2,e4,e7,e9}	{e1,e5,e6}
e4	{e2,e5,e7,e8}	{e1,e2,e6}
e5	{e4,e6,e8,e9}	{e1,e2,e3}
e6	{e1,e5,e7,e9}	{e2,e3,e4}
e7	{e1,e3,e4,e6}	{e2,e5}
e8	{e1,e2,e4,e5}	{e3,e6}
e9	{e2,e3,e5,e6}	{e1,e4}

Neighborhood and shortest path algorithm in weighted graph:

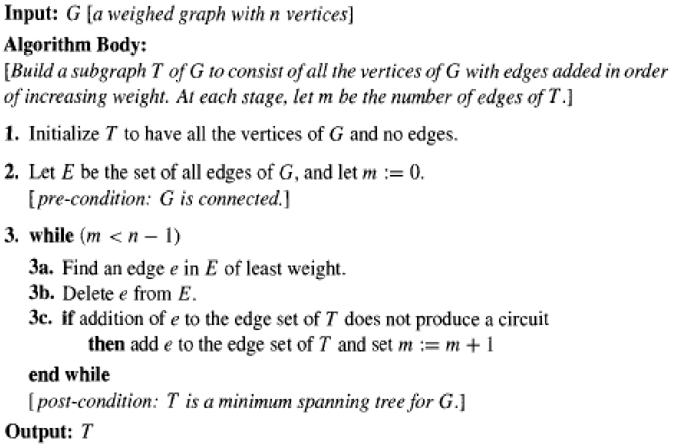
Definition 1:

Shortest path algorithm for n-neighborhood for a vertex v on graph G is the n-neighborhood of a vertex that have the smallest weight.

Note:

To find shortest path algorithm for n-neighborhood we must optain minimum spanning tree first by using appropriate algorithm.

Kruskal's Algorithm[5]



```
Input: G [a weighed graph with n vertices]
Algorithm Body:
[Build a subgraph T of G to consist of all the vertices of G with edges added in order
of increasing weight. At each stage, let m be the number of edges of T.]
1. Initialize T to have all the vertices of G and no edges.
2. Let E be the set of all edges of G, and let m := 0.
   [pre-condition: G is connected.]
3. while (m < n − 1)
    3a. Find an edge e in E of least weight.
    3b. Delete e from E.
    3c. if addition of e to the edge set of T does not produce a circuit
            then add e to the edge set of T and set m := m + 1
    end while
    [post-condition: T is a minimum spanning tree for G.]
Output: T
```

Example 11:

Describe the action of Kruskal's algorithm for the graph shown in Figure 11

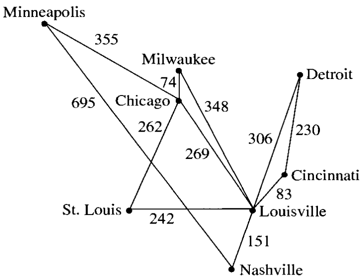
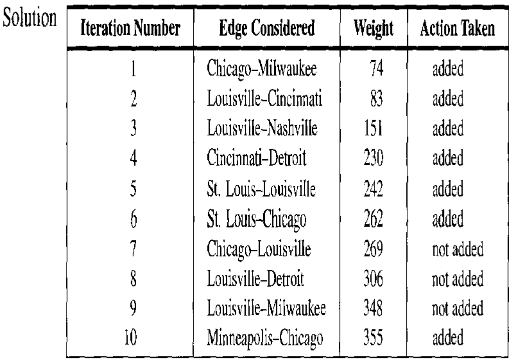


Figure (11)

Solution



Iteration Number	Edge Considered	Weight	Action Taken
1	Chicago–Milwaukee	74	added
2	Louisville–Cincinnati	83	added
3	Louisville–Nashville	151	added
4	Cincinnati–Detroit	230	added
5	St. Louis–Louisville	242	added
6	St. Louis–Chicago	262	added
7	Chicago–Louisville	269	not added
8	Louisville–Detroit	306	not added
9	Louisville–Milwaukee	348	not added
10	Minneapolis–Chicago	355	added

The tree produced by Kruskal's algorithm is shown in Figure 12

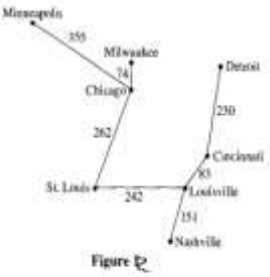


Figure 12

Then we can determine first and second neighborhood as follows:

vertex	$N^1(v)$	$N^2(v)$
Minneapolis	{Chicago}	non
Chicago	{Milwaukee }	{ Louisvill}
Milwaukee	{ Chicago}	{St.Louis}
Detroit	{ Cincinnati }	{ Louisvill}
Cincinnati	{ Louisvill}	{ Nashvill }
Louisvill	{ Cincinnati }	{ Detroit}
Nashvill	{ Louisvill}	{ Cincinnati }
St.Louis	{ Louisvill}	{ Cincinnati }

Example 12:

For a graph shown in fig(13) ,compute first and second neighborhood of shortest path.

Solution:

First we find minimum spanning tree by describing the action of Kruskal's algorithm.

Iteration no.	Edge considered	weight	Action taken
1	DCA - JFK	370	added
2	YYZ - YUL	516	added
3	YUL - JFK	544	added
4	YYZ - JEK	593	not added
5	YYZ - LAX	3523	added
6	LAX - JFK	4010	not added

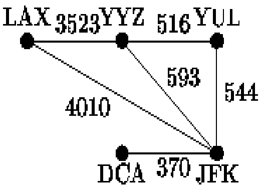


Fig (12)

the tree produced by Kruskal' s algorithm will be:

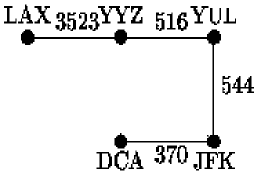


Fig (14)

then we can compute first and second neighborhood as follows:

vertex	$N^1(v)$	$N^2(v)$
LAX	{YYZ}	{YUL}
YYZ	{YUL}	{JFK}
YUL	{YYZ}	{DCA}
JFK	{DCA}	{YYZ}
DCA	{JFK}	{YUL}

Reference:

[1] Bondy, John Adrian; Murty, U. S. R. (1976), Graph Theory with Applications, North-Holland, ISBN 0-444-19451-7, page 5.

[2] Fournier,Jean-Claude,Graph Theory and applications with Exercises and problems, ISTE Ltd,2009.

[3]Grimadi Ralph,P., Discrete and combinatorial mathematics:An Applied Introduction, Fifth Edition, Pearson Education, Inc.,2004

[4] Pemmaraju, S. and Skiena, S. "Cycles, Stars, and Wheels." §6.2.4 in computational Discrete Mathematics combinatiorics and graph theory in mathematica Cambridge, England: Cambridge University Press, pp. 248-249, 2003.

[5]Susanna S.Epp, Discrete Mathematics With Application, Third Edition, Thomson Learning,Inc,2004.