New algorithm for graph with graphs vertices
EL-Zohny.H, Salam. R and EL-Morsy.H
Department of Mathematics, Faculty of Science Al-Azahar University, Cairo, Egypt.

ABSTRACT
In this paper we will compute a new algorithm for new graph which its vertex is a graph.

Keywords
Algorithm ,
Graphs vertices.

Input:
Null graph with graphs vertices \( V^n \) , \( V^m \) where n is outer vertices , m is its internal vertices.

Algorithm body:
1. Create a subgraph that visit each outer vertices \( V^n \) and its internal vertices \( V^m \).
2. initialized T to have all vertices of G "which have outer vertices."
3. select the smallest superscript \( k \) for \( 1 \leq k \leq i , 0 \leq n \leq j \).
4. \( V^n m^k \) has not already been visited.
5. If no superscript is found , then,
6. Go to step 3 , otherwise,
7. Perform the following:
8. 2a. attach the internal edge \( \{ V^1m^k , V^n e \} \) to T, and visit \( V^1m^k \).
9. 2b. assign \( V^1m^k \) to \( V^n m^k \) and ,
10. 2c. return to step 2.
11. End while.
12. Output T .
end algorithm.

Example 1:
For a null graph shown in fig(1) we can compute its algorithm as follows:

Spanning tree for a graph G is a subgraph of G that contains every vertex of G and is a tree[4].

Main results:
We will discuss two new algorithms for graph with graphs vertices.
1. algorithm for null graph which vertices is a graph:
Input:
Null graph with graphs vertices \( V^{0m} \), \( 0 \leq n \leq 3 \), \( 0 \leq m \leq 5 \).

Algorithm body:
Create a subgraph that visit each outer vertices \( V^o \) then its internal vertices \( V^{imi} \).
Proceeding from vertex to vertex but moving along internal spanning tree of that graph.
1. initialized \( T \) to have all vertices of \( G \) "which have outer vertices".
2. select \( V^o \) and visit all internal vertices \( V^{imi} \) to \( V^{imi} \).
2a. attach internal edges \{ \( V^{imi},V^{imi} \) \}, \( \ldots \\ldots \{ \( V^{imi},V^{imi} \) \} \) to \( T \).
2b. go to step 2 for the other vertices \( V^o \).
3. output \( T \).
4. output \( T',T'' \).
5. end algorithm.

Algorithm for which vertices is graph:
Input:
Connected graph \( G(V,E) \), \( V(G) = \{V^o,V^i\} \), \( V^o = \{\{V^{00},V^{01}\}\} \), \( V^i = \{\{V^{0},V^{1}\}\} \).

Algorithm body:
Create a subgraph that visit each outer vertices \( V^o \) then its internal vertices \( V^{imi} \).
Proceeding from vertex to vertex but moving along internal spanning tree \( T \) of that graph, then along it's outer spanning tree \( T' \).
1. initialized \( T \) to have vertex \( v^o \).
2. let \( E \) the set of all edges of \( G \), \( m = 0 \).
3. while \( (m \leq 1) \).
3a. visit outer vertex \( v^o \) then visit \( V^{00},V^{01} \).
3b. attach the internal edge \( \{V^{00},V^{01}\} \) to \( T \).
3c. attach the outer edge \( e \) to \( T \) and visit \( V^o \).
3d. return to step 3.
End while.
4. output \( T,T' \).
end algorithm.

Example 3:
For a graph shown in fig(3) we have:
Input:
Connected graph \( G(V,E) \), \( V(G) = \{V^o,V^i\} \), \( V^o = \{\{V^{00},V^{01}\}\} \), \( V^i = \{\{V^{0},V^{1}\}\} \).

Algorithm body:
Create a subgraph that visit each outer vertices \( V^o \) then its internal vertices \( V^{imi} \).
Proceeding from vertex to vertex but moving along internal spanning tree \( T \) of that graph, then along it’s outer spanning tree \( T' \).
1. initialized \( T \) to have vertex \( v^o \).
2. let \( E \) the set of all edges of \( G \), \( m = 0 \).
3. while \( (m \leq 2) \).
3a. visit outer vertex \( v^o \) then visit \( V^{00},V^{01},V^{02} \).
3b. attach internal vertices \( \{V^{00},V^{01}\},\{V^{01},V^{02}\},\{V^{02},V^{01}\}\) to \( T \).
3c. attach the outer edge $e'$ to $T$ and visit $V'$.
3d. return to step 3 for the other vertices.
End while.
4. output $T,T'$.
end algorithm.

**Example 4:**
For a graph shown in fig(4).

![Fig(4)](image)

**Input:**
Connected graph $G(V, E)$, $V(G) = \{V_0, V_1, V_2\}$, $V_0 = \{V_{00}, V_{01}, V_{02}\}$, $V_1 = \{V_{10}, V_{11}, V_{12}\}$, $V_2 = \{V_{20}, V_{21}, V_{22}\}$.

**Algorithm body:**
Create a subgraph that visit each outer vertices $V$ then its internal vertices $V_{im}$.

Proceeding from vertex to vertex but moving along internal spanning tree $T$ of that graph, then along it's outer spanning tree $T'$.
1. initialized $T$ to have vertex $V_0$.
2. let $E$ the set of all edges of $G$, $m = 0$.
3. while ($m \leq 2$).
3a. visit outer vertex $V_0$.
then visit $V_00, V_01, V_02$.
3b. attach internal vertices $\{V_{00}, V_{01}, V_{02}\}$ to $T$.
3c. attach the outer edge $e'$ to $T'$ and visit $V_1$.
3d. return to step 3 for the other vertices.
End while.
4. output $T, T'$.
end algorithm.

**References:**