**DFS ( Depth-first search ) אלגוריתם חיפוש לעומק**

Vertex Stack

1/

**start - vertex 0**

**0**

Vertex Stack

1/ 2/

**1**

**0**

Vertex Stack

1/ 2/ 3/

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

5/

**6**

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

6/ 5/

**4**

**6**

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

6/7 5/

8/

🡪 **4**

**6**

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

6/7 5/

8/

**5**

**6**

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

6/7 5/

8/9

🡪 **5**

**6**

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

6/7 5/ 10/

8/9

**7**

**6**

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

6/7 5/12 10/11

8/9

🡪 **7**

**6**

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/

6/7 5/12 10/11

8/9

🡪 **6**

**3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/ 4/13

6/7 5/12 10/11

8/9

**🡪 3**

**2**

**1**

**0**

Vertex Stack

1/ 2/ 3/14 4/13

6/7 5/12 10/11

8/9

* **2**

**1**

**0**

Vertex Stack

1/ 2/15 3/14 4/13

6/7 5/12 10/11

8/9

* **1**

**0**

16/

1/ 2/15 3/14 4/13

6/7 5/12 10/11

8/9

**8**

**0**

Vertex Stack

16/17

1/ 2/15 3/14 4/13

6/7 5/12 10/11

8/9

* **8**

**0**

Vertex Stack

16/17

1/18 2/15 3/14 4/13

6/7 5/12 10/11

8/9

🡪 **0**

Vertex Stack is **empty**

**DFS Iterative JAVA code**

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Iterator;

**import** java.util.Stack;

**public** **class** DFS\_Loop2 {

**public** **static** **void** dfs(ArrayList<Integer>[] g, **int** s) {

**int** n = g.length;

**boolean**[] marked = **new** **boolean**[n];

**int** timeFirst[] = **new** **int**[n];

**int** timeLast[] = **new** **int**[n];

**int**[] pred = **new** **int**[n];

**int** nil = -1;

Stack<Integer> st = **new** Stack<Integer>();

**for** (**int** i = 0; i < n; i++) {

timeFirst[i] = 0;

timeLast[i] = 0;

pred[i] = nil;

marked[i] = **false**;

}

**int** time = 0;

Iterator<Integer>[] adj = (Iterator<Integer>[]) **new** Iterator[n];

**for** (**int** j = 0; j < adj.length; j++) {

adj[j] = g[j].iterator();

}

pred[s] = nil;

marked[s] = **true**;

st.push(s);

timeFirst[s] = ++time;

System.***out***.println(st);

**while**(!st.isEmpty()){

**int** u = st.peek();

**if**(adj[u].hasNext()){

**int** v = adj[u].next();

**if** (!marked[v]){

marked[v] = **true**;

pred[v] = u;

timeFirst[v] = ++time;

st.push(v);

System.***out***.println(st);

}

}

**else**{

st.pop();

timeLast[u] = ++time;

System.***out***.println(st);

}

}

System.***out***.println("pred: "+Arrays.*toString*(pred));

System.***out***.println("isVisited: "+Arrays.*toString*(marked));

System.***out***.println("timeFirst: "+Arrays.*toString*(timeFirst));

System.***out***.println("timeLast: "+Arrays.*toString*(timeLast));

}

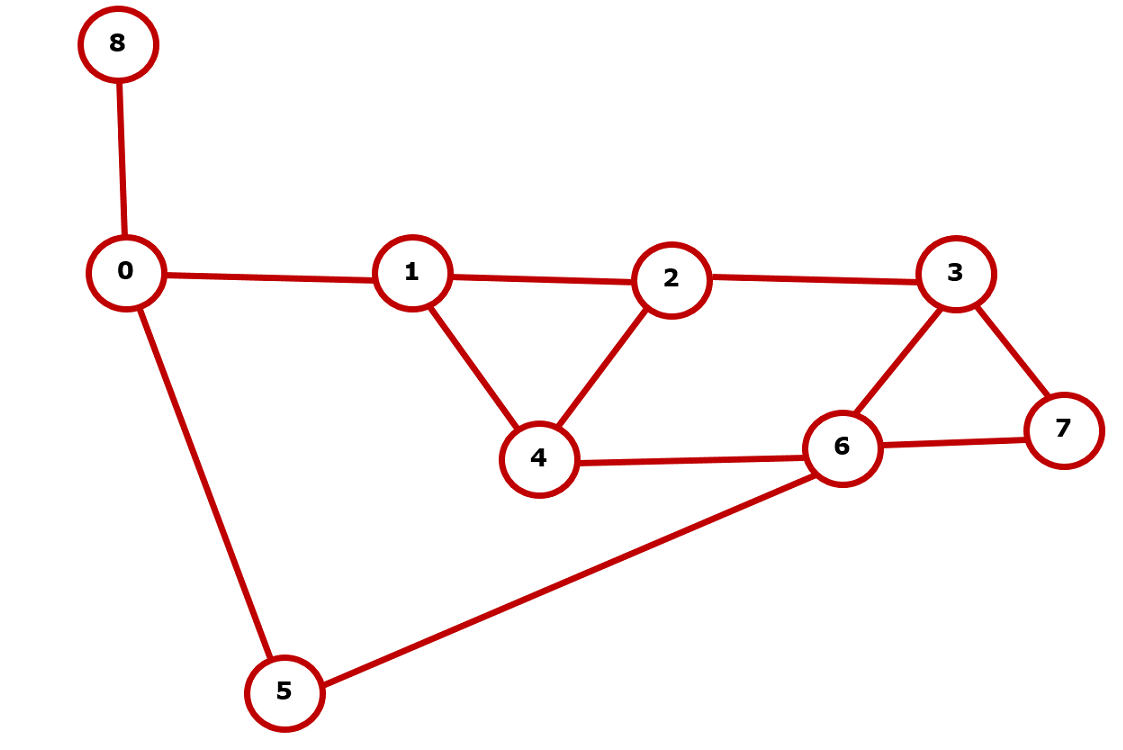
**public** **static** **void** main(String[] args) {

DFS\_Loop2.*dfs*(GraphArrayListInit.*initGraph2*(), 0);

}

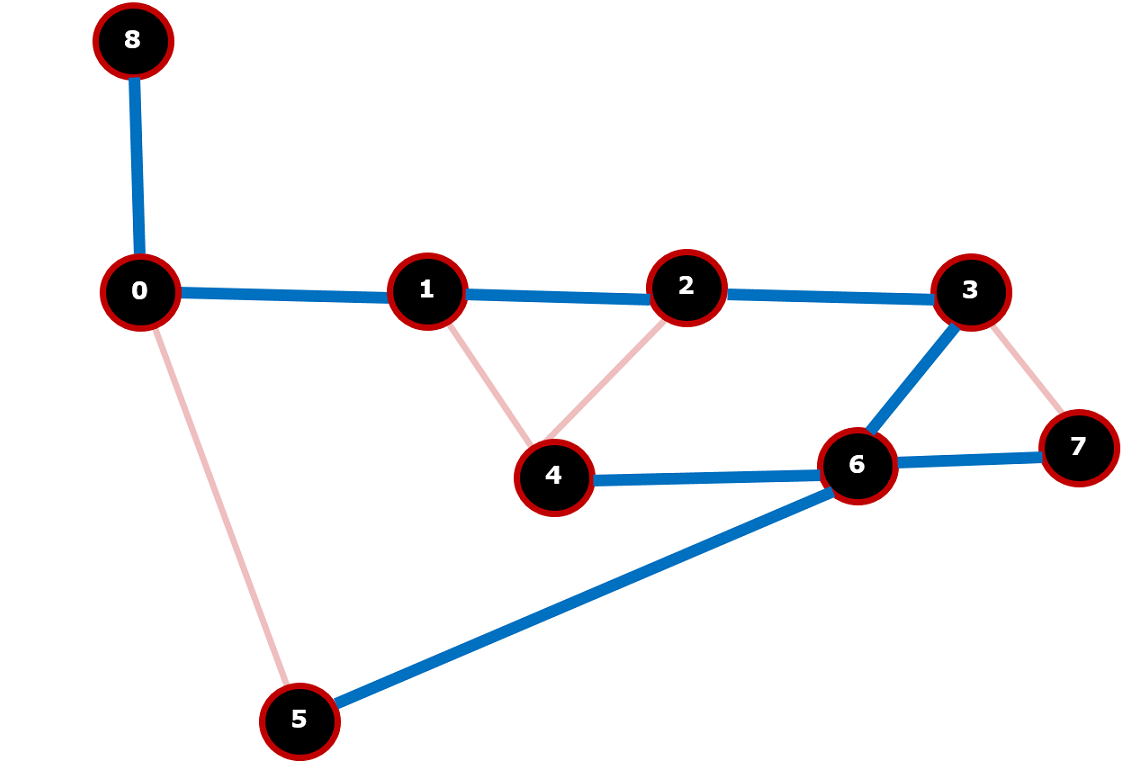
}

**start - vertex 0**

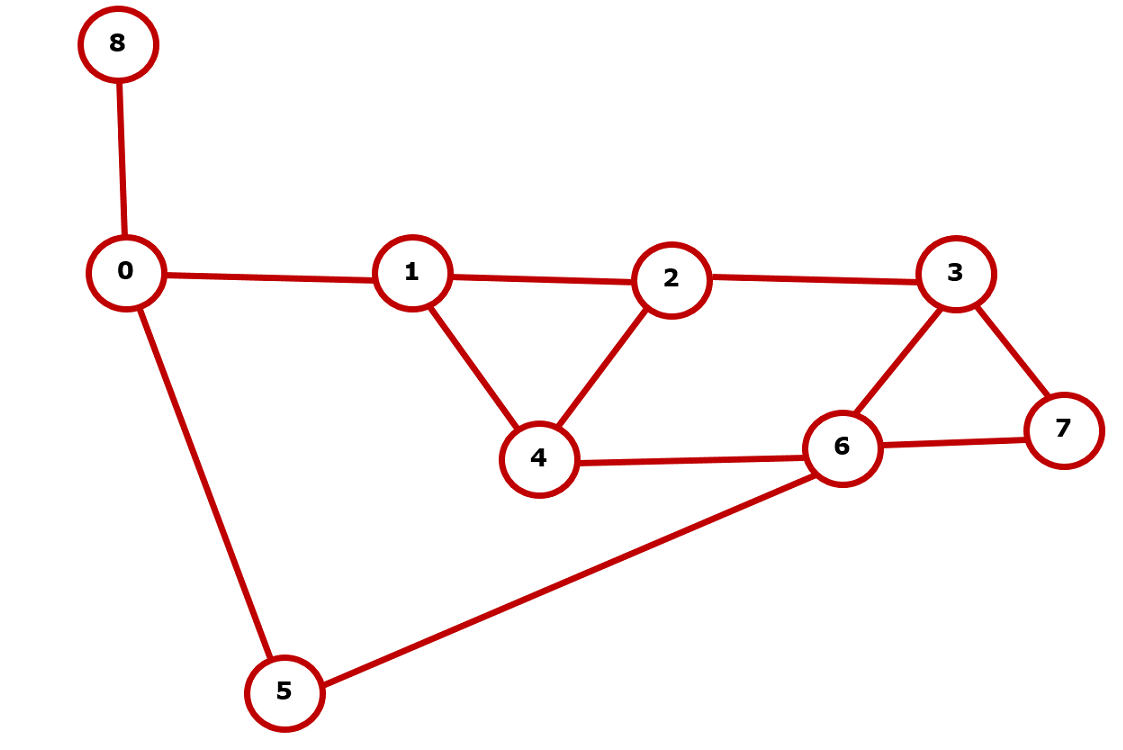
****

**name : 0, 1, 2, 3, 4, 5, 6, 7, 8**

**parent : -1, 0, 1, 2, 6, 6, 3, 6, 0**

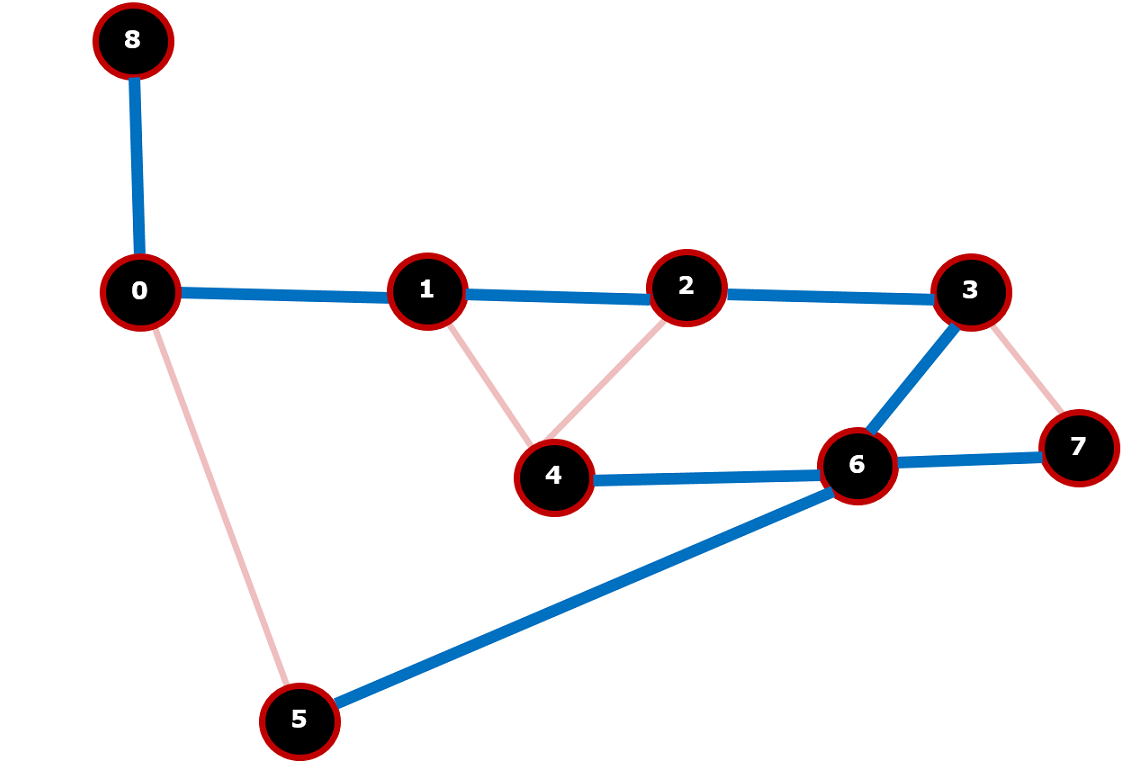
****

**start - vertex 3**

****

**name : 0, 1, 2, 3, 4, 5, 6, 7, 8**

**parent : 1, 2, 3, -1, 6, 0, 5, 6, 0**

****