בדיקה האם גרף בעל רכיב קשירות אחד

אם מטריצה בעלת רכיב קשירות אחד אז כל האיבריםשל מטריצה חייבים להיות ,TRUE אחרת יש קודקוד שאין מסלולים בין קודקודים אחרים לקודקוד הזה.

זה אומר שבבדיקת מטריצה בוליאנית, FALSE ראשוןנותן תשובה שלילית לשאלה.

מספיק לבדוק שורה ראשונה במטריצה.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **static** **boolean**[][] initB(){

**boolean** [][] mat = {

{**false**,**false**,**false**,**false**,**true**,**false**,**false**,**false**,**false**}, // 0

{**false**,**false**,**false**,**false**,**false**,**false**,**true**,**false**,**false**}, // 1

{**false**,**false**,**false**,**false**,**false**,**true**,**false**,**true**,**false**}, // 2

{**false**,**false**,**false**,**false**,**false**,**false**,**false**,**false**,**false**}, // 3

{**true**,**false**,**false**,**false**,**false**,**false**,**false**,**false**,**true**}, // 4

{**false**,**false**,**true**,**false**,**false**,**false**,**false**,**true**,**false**}, // 5

{**false**,**true**,**false**,**false**,**false**,**false**,**false**,**false**,**false**}, // 6

{**false**,**false**,**true**,**false**,**false**,**true**,**false**,**false**,**false**}, // 7

{**false**,**false**,**false**,**false**,**true**,**false**,**false**,**false**,**false**}}; // 8

**return** mat;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **static** **boolean**[][] initB1(){

**boolean** [][] mat = {

{**false**,**true**,**true**,**false**},

{**true**,**false**,**false**,**true**},

{**true**,**false**,**false**,**true**},

{**false**,**true**,**true**,**false**}};

**return** mat;

}

}

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

**boolean**[][] mat = *initB*();

//boolean[][] mat = initB1();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Problem 1

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\* Problem 1 \*\*\*\*\*\*\*\*\*\*\n");

*FloydWarshallBoolean*(mat);

*printBoolMatrix*(mat);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Problem 4

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\* Problem 4 \*\*\*\*\*\*\*\*\*\*\n");

System.***out***.println("is the graph connected? " + *isConnectedComplexN*(mat));

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\* Problem 4 \*\*\*\*\*\*\*\*\*\*\n");

System.***out***.println("is the graph connected? " + *isConnected*(mat));

}

**פלט:**

**boolean**[][] mat = *initB*();

\*\*\*\*\*\*\*\*\*\* Problem 4 \*\*\*\*\*\*\*\*\*\*

is the graph connected? false

\*\*\*\*\*\*\*\*\*\* Problem 4 \*\*\*\*\*\*\*\*\*\*

is the graph connected? False

**boolean**[][] mat = *initB1*();

\*\*\*\*\*\*\*\*\*\* Problem 4 \*\*\*\*\*\*\*\*\*\*

is the graph connected? true

\*\*\*\*\*\*\*\*\*\* Problem 4 \*\*\*\*\*\*\*\*\*\*

is the graph connected? true

**משימה:**

לבדוק האם גרף קשיר (בעל רכיב הקשירות אחד) –

* ליצור פונקציה שבודקת האם גרף קשיר בסיבוכיות

**public** **static** **boolean** isConnected(**boolean** mat[][])//O(n^2)

* ולאחר מכך לשפר את היעילות ל- - ליצור פונקציה נוספת שבודקת האם גרף קשיר בסיבוכיות

**public** **static** **boolean** isConnectedComplexN(**boolean** mat[][])//O(n)