

Tarea 5, para el viernes 13 de octubre

Ejercicio 1. Conectividad en grafos (40 puntos)

Dado un grafo posiblemente no conexo, obtener los subconjuntos conexos del grafo, es decir, se deben obtener cuales nodos pertenecen a cada subconjunto.

Implemente un método que determine si un grafo es conexo o no (que mande llamar el mínimo de veces a la búsqueda en profundidad). Este método imprime T o F en caso de ser o no conexo y los subconjuntos conexos en cada línea.

Ejemplo: sea el grafo dado por los siguientes aristas (A,B), (B,C), (D,E) el resultado será falso e imprimirá lo siguiente:

```
F
ABC
DE
```

Indicando que es un grafo no conexo con 2 partes las de los nodos A,B,C y la otra parte con D,E

Entrada:

```
5 3 //EL 5 indica que hay 5 nodos y 3 que se enlistaran a continuación 3 aristas (3 renglones)
A B //indica que A y B están conectados
B C
D E
```

SALIDA

```
F // F de false porque no
ABC // ABC (ordenados y sin espacio),
DE // DE (ordenados)
```

Ejercicio 2: CAMINO DFS (60 puntos)

Dado un grafo conexo, se desea obtener el camino para llegar del nodo A al nodo B usando la búsqueda en profundidad pero solamente queremos saber el camino que lleva directamente al nodo deseado. Es decir no deben mostrarse las otras rutas que no llegan al nodo que se busca. Modifique el DFS visto en clase para que muestre la ruta.

```
boolean t = true, f = false;

boolean[][] graph2 = {{ f, t, t, t, f, f, f },
{ t, f, t, f, t, f, f },
{ t, t, f, t, t, f, f },
{ t, f, t, f, f, f, t },
{ f, t, t, f, f, t, f },
{ f, f, f, f, t, f, f },
{ f, f, f, t, f, f, f }
};
```

Para llegar del nodo A al nodo B

El DFS muestra: A D G C E F B

El esperado es: A D C E B

Entrada (la entrada no incluye comentarios)

```
4 3 //NUM NODOS Y NUM ARISTAS
A D //ORIGEN Y DESTINO
A B //ARISTA 1
B C //ARISTA 2
C D //ARISTA 3
```

Salida

ABCD

Ejercicio 3: (35 puntos)

El dueño de facebook te pide que le ayudes a realizar un análisis de su red social. Dice que hay algunos usuarios que buscan a varios conocidos a la vez pero que al final de cuentas se quedan platicando con el que llega en menos brincos (en una búsqueda de anchura) y si ambos están en el mismo número de brincos se quedan con el que sea primero respecto de su nombre. Te pide que le programes un método que regrese al primero de sus conocidos.

Ejemplo:

Se muestran los vecinos de cada nodo

A:{B}
B:{A,E}
C:{D,E}
D:{C,F}
E:{B,C,F}
F:{D,E}

El usuario A busca a C, D, F

En la búsqueda en anchura se realiza el siguiente recorrido: A,B,E,C,F,D por lo que el primero en encontrar es el usuario C. esta es la respuesta.

Entrada

6 6 3//NUM NODOS, NUM ARISTAS, NUM OBJETIVOS

A C D F//NODO INICIAL, OBJETIVOS

A B

B E

E C

E F

C D

F D

Salida

C