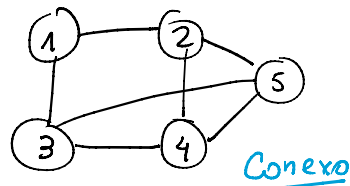


Un grafo es **conexo** si entre cada dos nodos hay un camino.

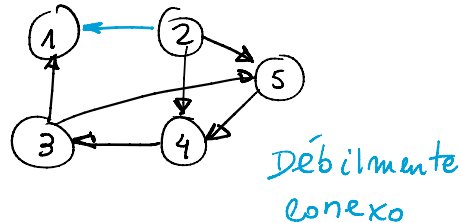
Un grafo no dirigido es **conexo** si existe un camino desde cualquier vértice a cualquier otro.

Un grafo dirigido con esta propiedad se denomina **fuertemente conexo**.

Si un grafo dirigido no es fuertemente conexo, pero el grafo subyacente (sin sentido en los arcos) es conexo, el grafo es **débilmente conexo**.



Conexo

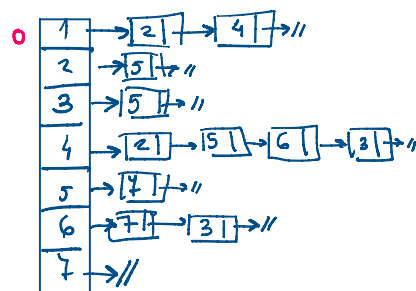
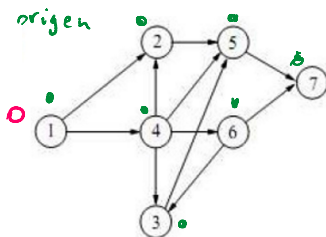


Débilmente conexo

dfs(origen):

1. Marcar origen como visitado
2. Para cada "adyacente" a origen:
 3. Si "adyacente" no está visitado:
 4. dfs(adyacente)

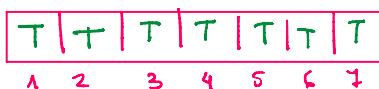
procesamos origen



Origen: ①

Adyacente:

Visitados:



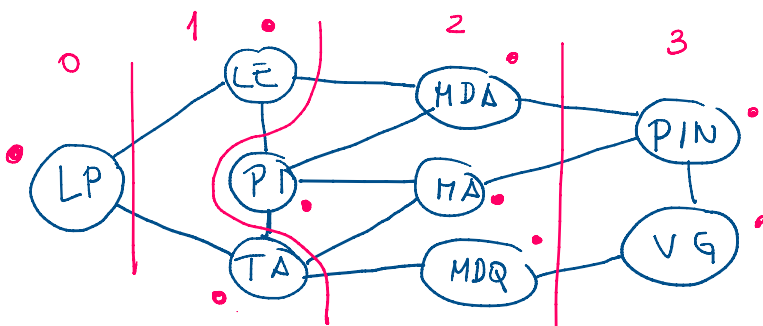
Recorrido:

1, 2, 5, 7, 4, 6, 3

```
Grafo <string> gr = new Grafo <string>();
Vertice <string> lp = new Vertice <string>("La Plata");
gr.agregarVertice(lp);
Vertice <string> pi = new Vertice <string>("Pilar");
gr.agregarVertice(pi);
Vertice <string> ta = new Vertice <string>("Tandil");
gr.agregarVertice(ta);
Vertice <string> ma = new Vertice <string>("Madariaga");
gr.agregarVertice(ma);
Vertice <string> le = new Vertice <string>("Lezama");
gr.agregarVertice(le);
Vertice <string> mda = new Vertice <string>("Mar de Ajo");
gr.agregarVertice(mda);
Vertice <string> mdq = new Vertice <string>("Mar del Plata");
gr.agregarVertice(mdq);
Vertice <string> vg = new Vertice <string>("Villa Gessel");
gr.agregarVertice(vg);
Vertice <string> pin = new Vertice <string>("Pinamar");
gr.agregarVertice(pin);
```

```
gr.conectar(lp, ta, 0);
gr.conectar(ta, lp, 0);
gr.conectar(lp, le, 0);
gr.conectar(le, lp, 0);
gr.conectar(le, pi, 0);
gr.conectar(pi, le, 0);
gr.conectar(le, mda, 0);
gr.conectar(mda, le, 0);
gr.conectar(ta, pi, 0);
gr.conectar(pi, ta, 0);
gr.conectar(ta, ma, 0);
gr.conectar(ma, ta, 0);
gr.conectar(mdq, ta, 0);
gr.conectar(ta, mdq, 0);
```

```
gr.conectar(pi, ma, 0);
gr.conectar(ma, pi, 0);
gr.conectar(pi, mda, 0);
gr.conectar(mda, pi, 0);
gr.conectar(mda, pin, 0);
gr.conectar(pin, mda, 0);
gr.conectar(mdq, vg, 0);
gr.conectar(vg, mdq, 0);
gr.conectar(pin, ma, 0);
gr.conectar(ma, pin, 0);
gr.conectar(pin, vg, 0);
gr.conectar(vg, pin, 0);
```

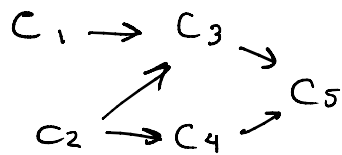


*** Recorrido DFS desde La Plata ***

La Plata Tandil Pilar Lezama Mar de Ajo Pinamar Madariaga Villa Gessel Mar del Plata

*** Recorrido BFS desde La Plata ***

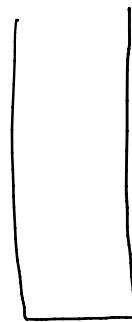
La Plata Tandil Lezama Pilar Madariaga Mar del Plata Mar de Ajo Pinamar Villa Gessel



grado-in

C_1	C_2	C_3	C_4	C_5
0	0	2	1	2
0	-	1	0	2
-	-	0	0	2
-	-	0	-	1
-	-	-	-	0
-	-	-	-	-

Sort: $C_2 C_1 C_4 C_3 C_5$



Pila

Sort:

$C_2 C_4 C_1 C_3 C_5$

$$\frac{n}{\log_2 n} \leq c \log_2 n$$

$$\begin{aligned}
 c &\geq \lim_{n \rightarrow \infty} \frac{n}{\log_2 n} = \lim_{n \rightarrow \infty} \frac{1}{\frac{2}{n} \log_2 n} = \lim_{n \rightarrow \infty} \frac{n}{\log_2 n} = \infty \\
 &= \lim_{n \rightarrow \infty} n = \infty
 \end{aligned}$$