

Estructuras de Datos II

RECORRIDOS EN GRAFOS

BÚSQUEDA EN ANCHURA BFS
(Breadth First Search)

BÚSQUEDA EN PROFUNDIDAD DFS
(Depth First Search)

BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Se comienza con un nodo u

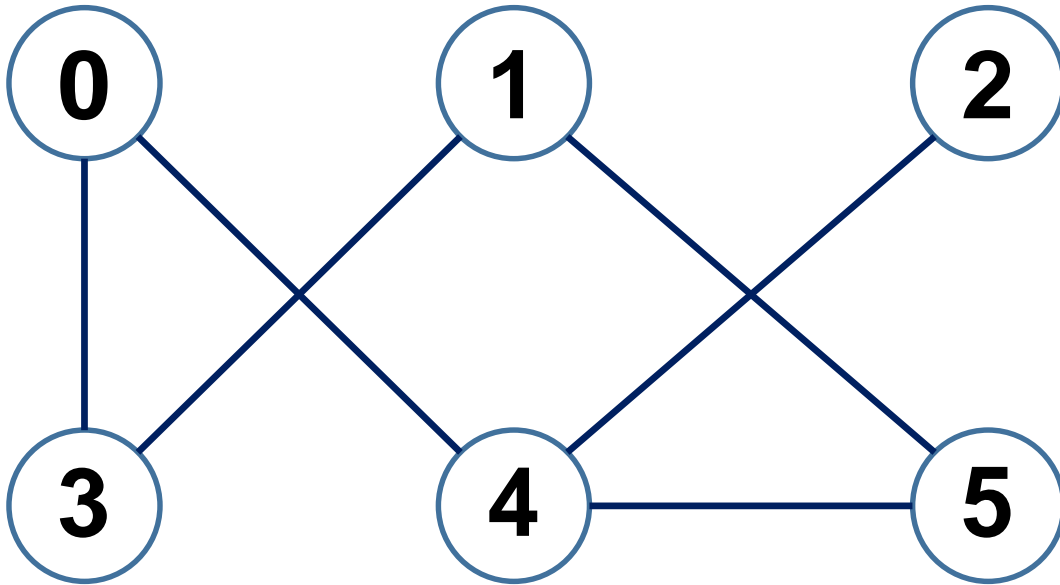
Se exploran todos los nodos v adyacentes al nodo u

Para cada uno de los nodos v , se exploran sus respectivos adyacentes **QUE NO HAYAN SIDO VISITADOS ANTES**

Se continúa de esta manera hasta recorrer **TODO** el grafo

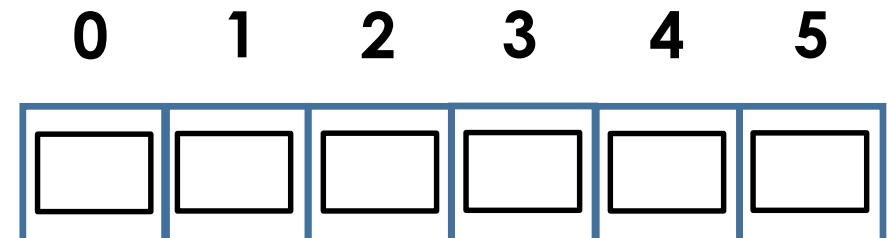
BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Nos apoyamos en una COLA para el recorrido por niveles



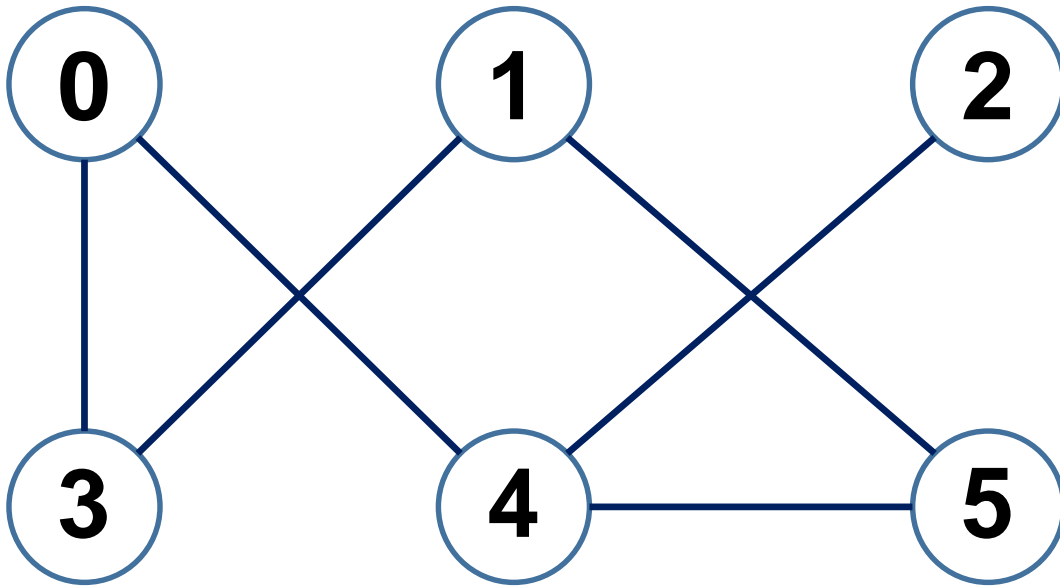
COLA
(QUEUE) →

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

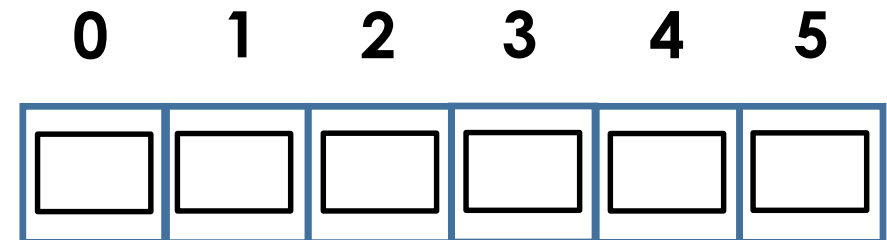
Podemos
empezar por
este nodo (0)



Nos apoyamos en una COLA
para el recorrido por niveles

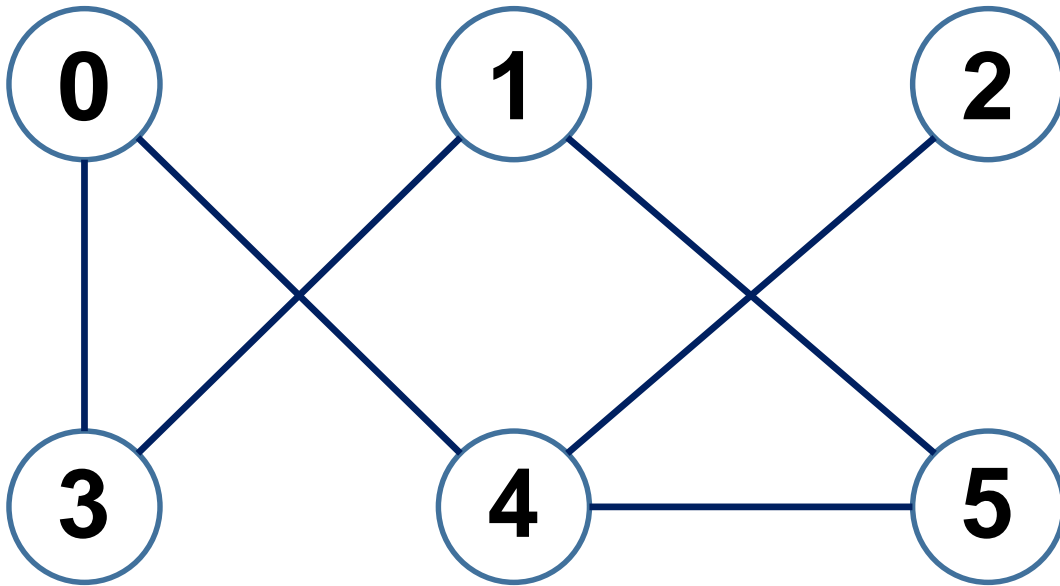
COLA
(QUEUE) →

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

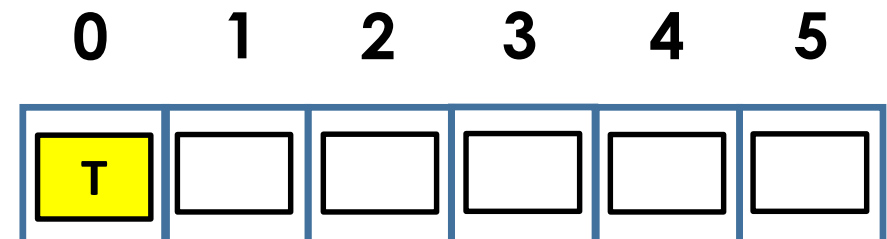


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) →

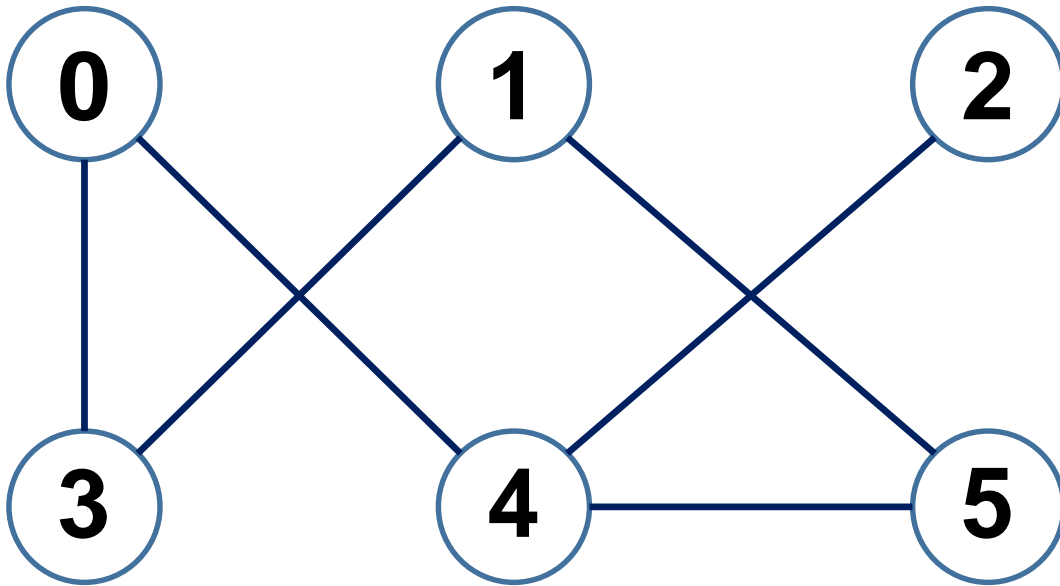
Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:

Lo marcamos como visitado



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)



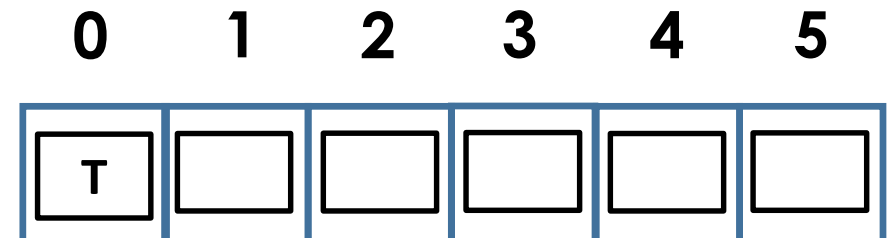
Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) →

0

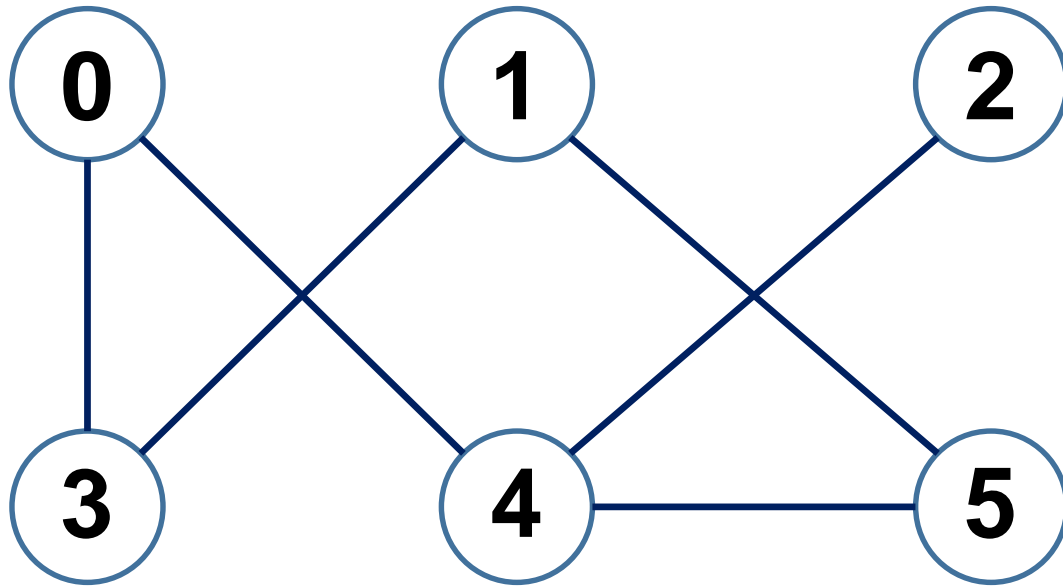
Lo insertamos en COLA

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)



Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE)

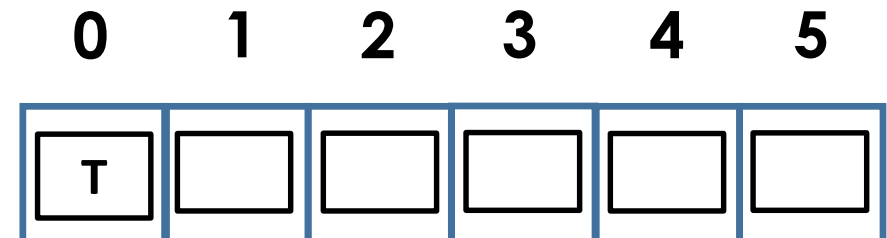
0

Lo insertamos en COLA

PROCEDIMIENTO:

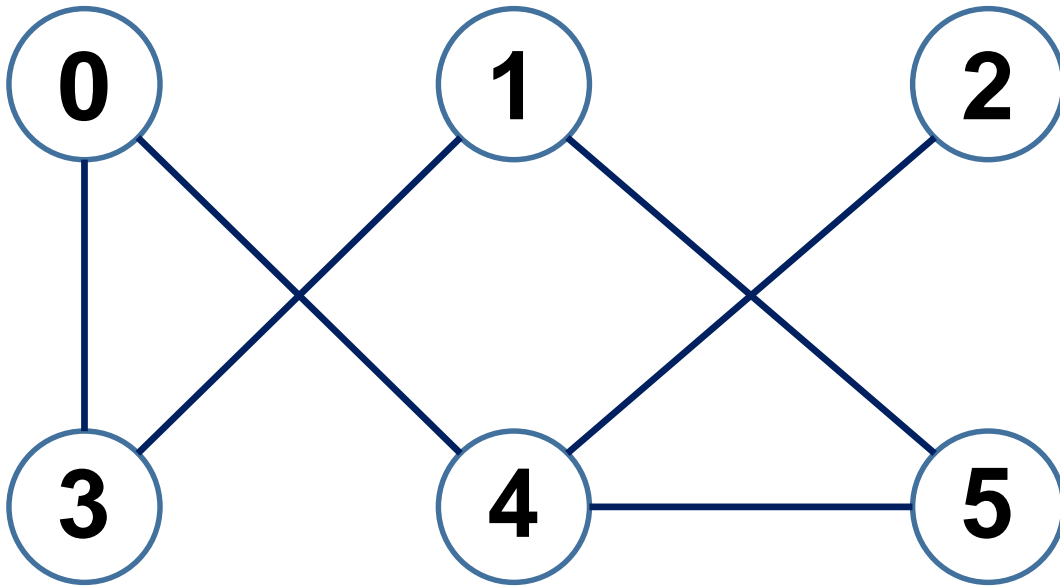
Mientras la cola NO esté vacía
tomamos cada nodo y
expandimos a sus adyacentes

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)



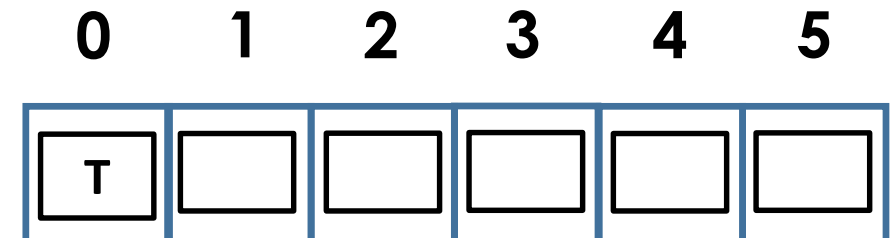
Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE)

0

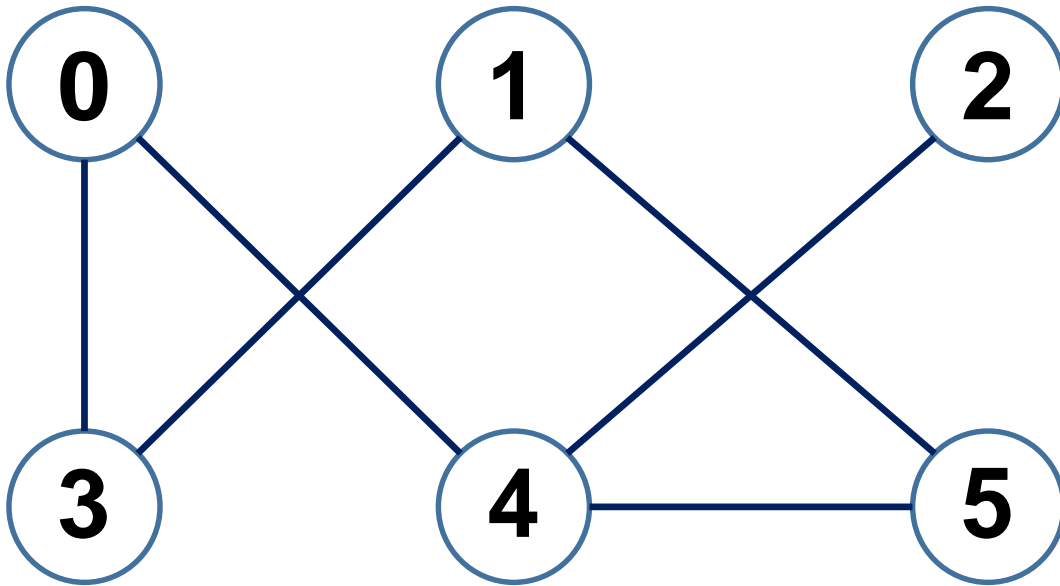
Adyacentes de 0: son 3 y 4.

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

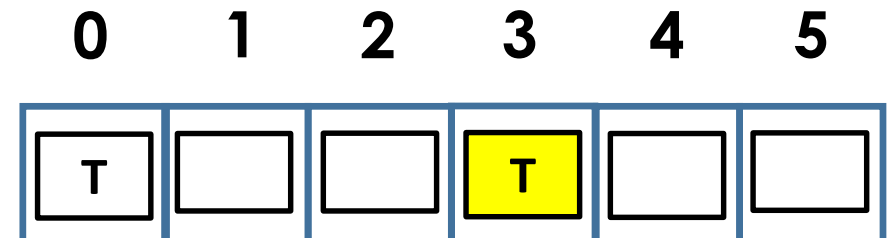


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) → 0

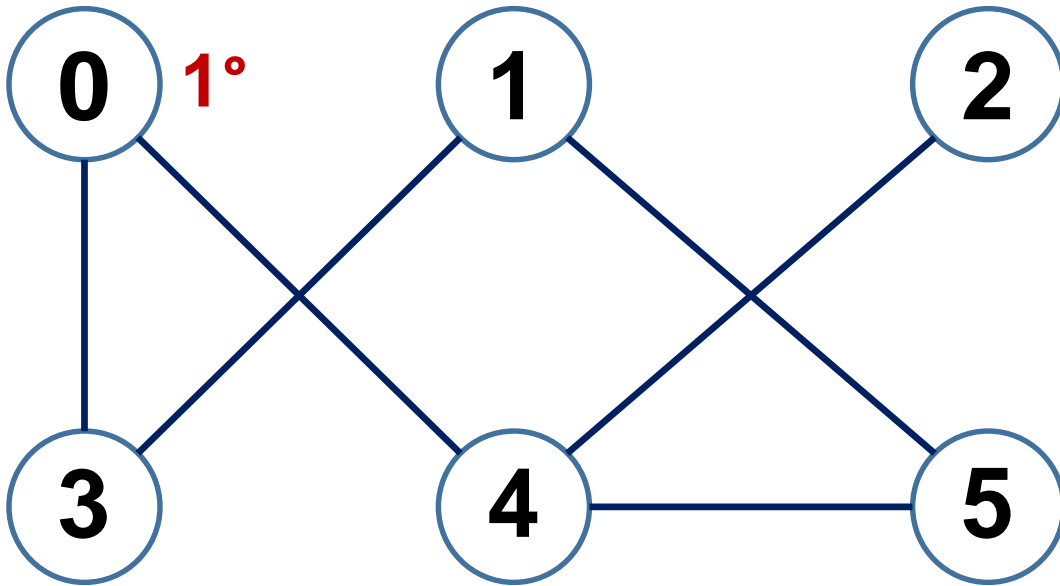
Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:

Marcamos a 3 como visitado



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

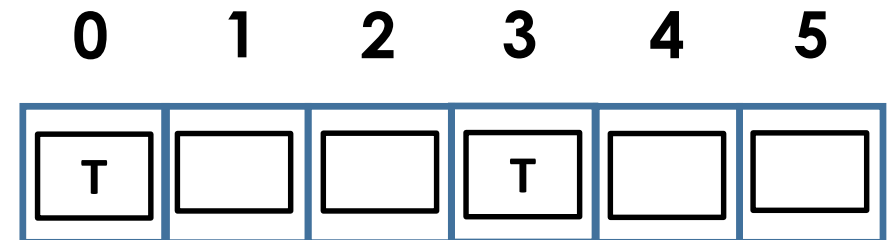


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) → 3

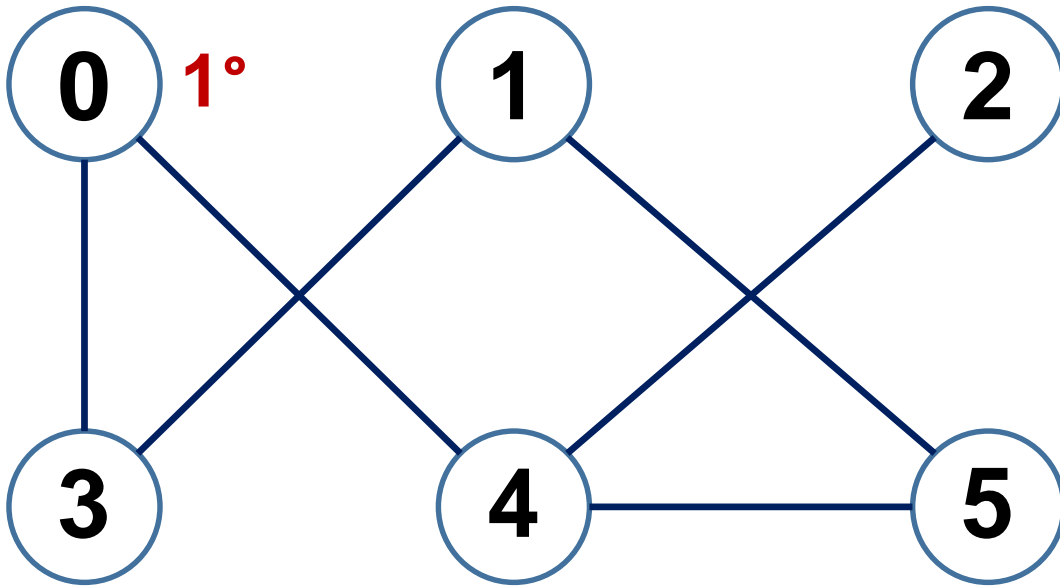
Se inserta en la COLA
(pero primero sacamos a 0
pues ya fue visitado)
Fue el 1° elemento visitado.

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

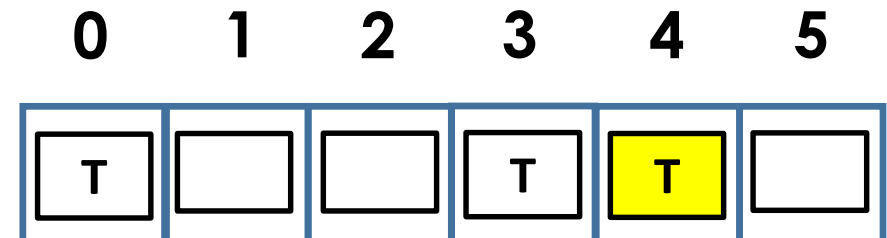


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) → 3

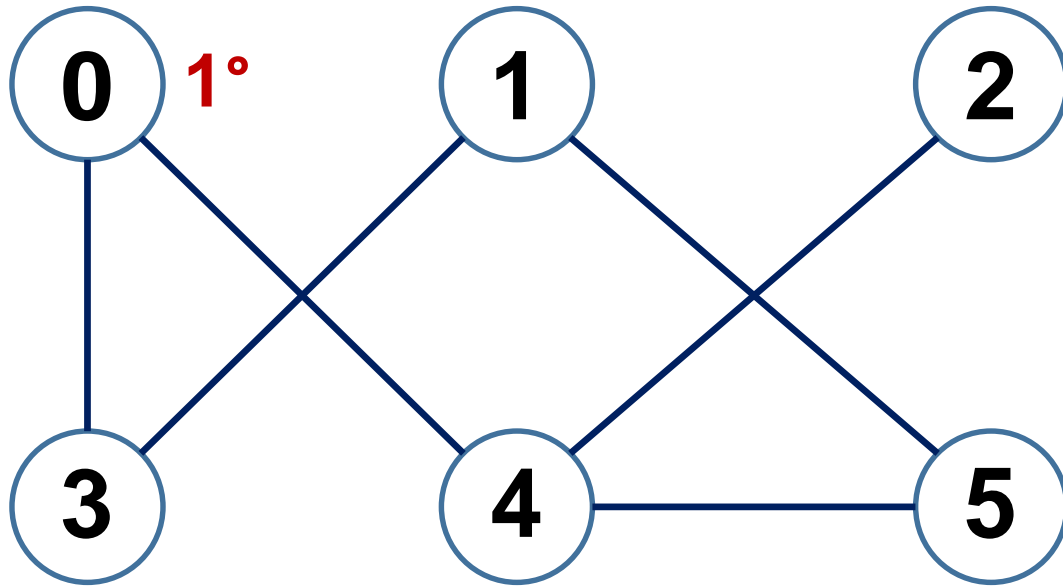
Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:

Marcamos a 4 como visitado



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)



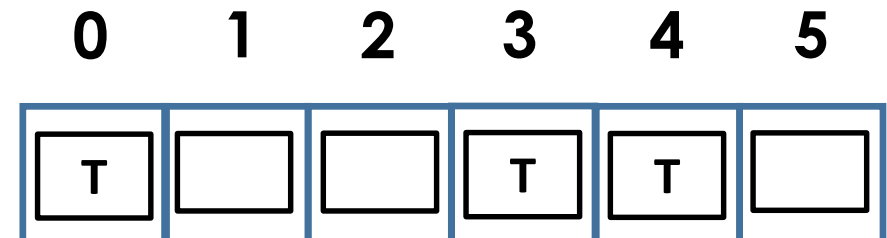
Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE)

3
4

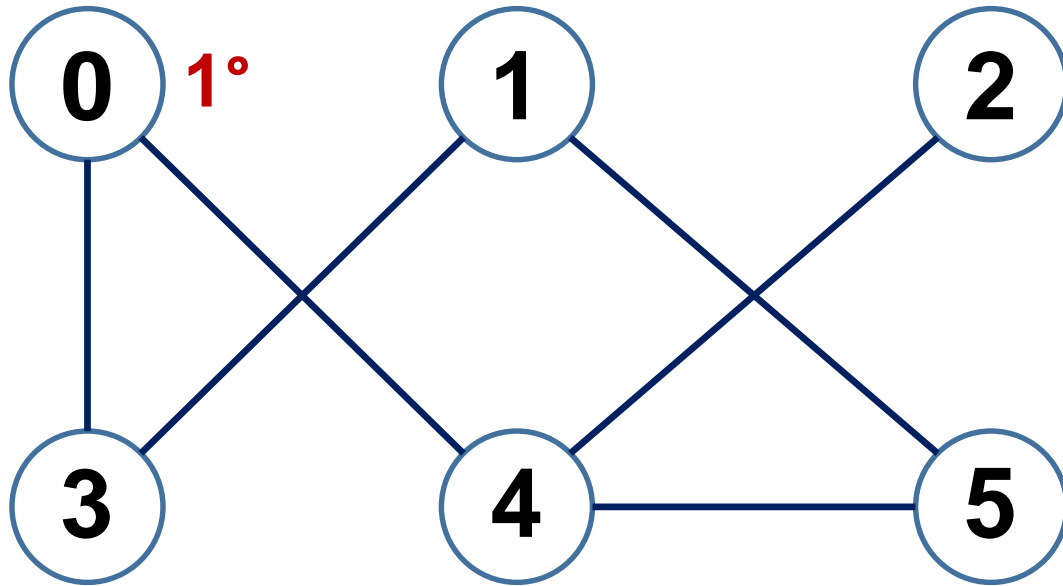
Se inserta en la COLA

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:

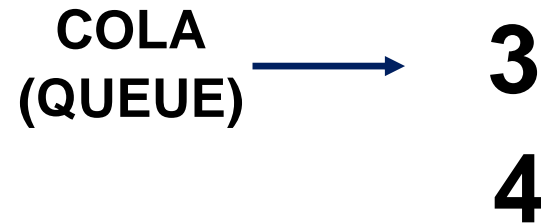


BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

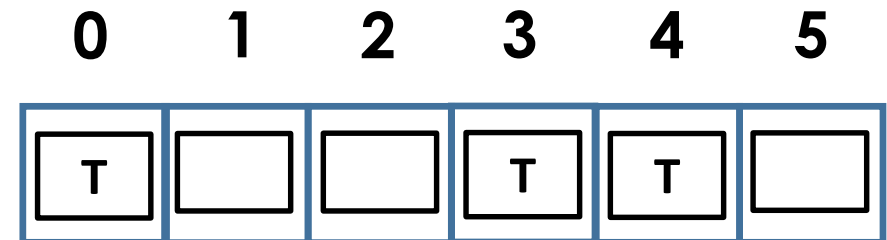


Nos apoyamos en una COLA
para el recorrido por niveles



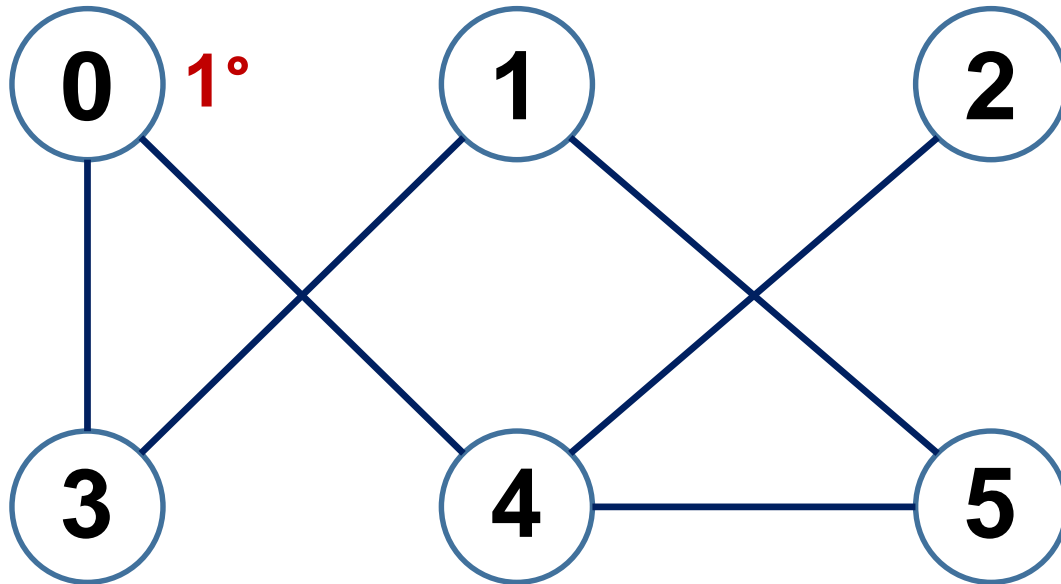
Cola NO está vacía.
Expandimos a los adyacentes de 3
Que son 0 y 1

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:

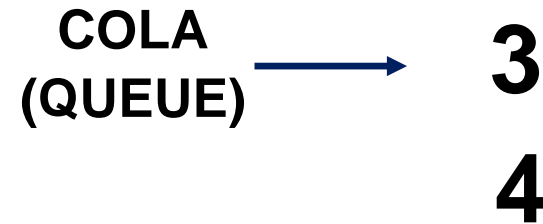


BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

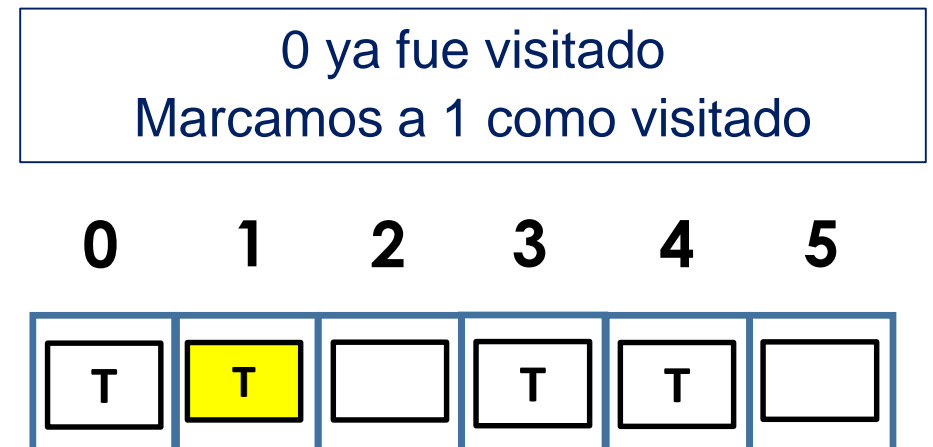
Podemos
empezar por
este nodo (0)



Nos apoyamos en una COLA
para el recorrido por niveles

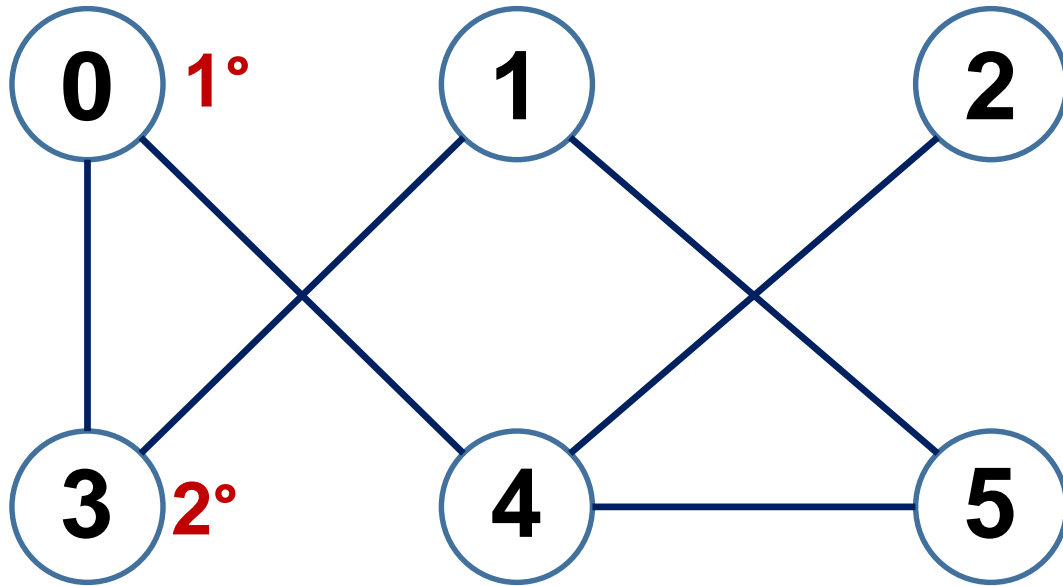


Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

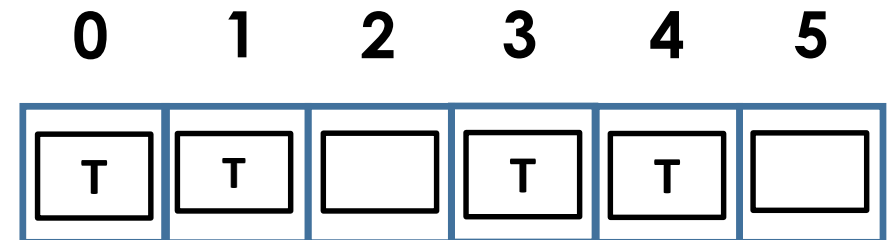


Nos apoyamos en una COLA
para el recorrido por niveles

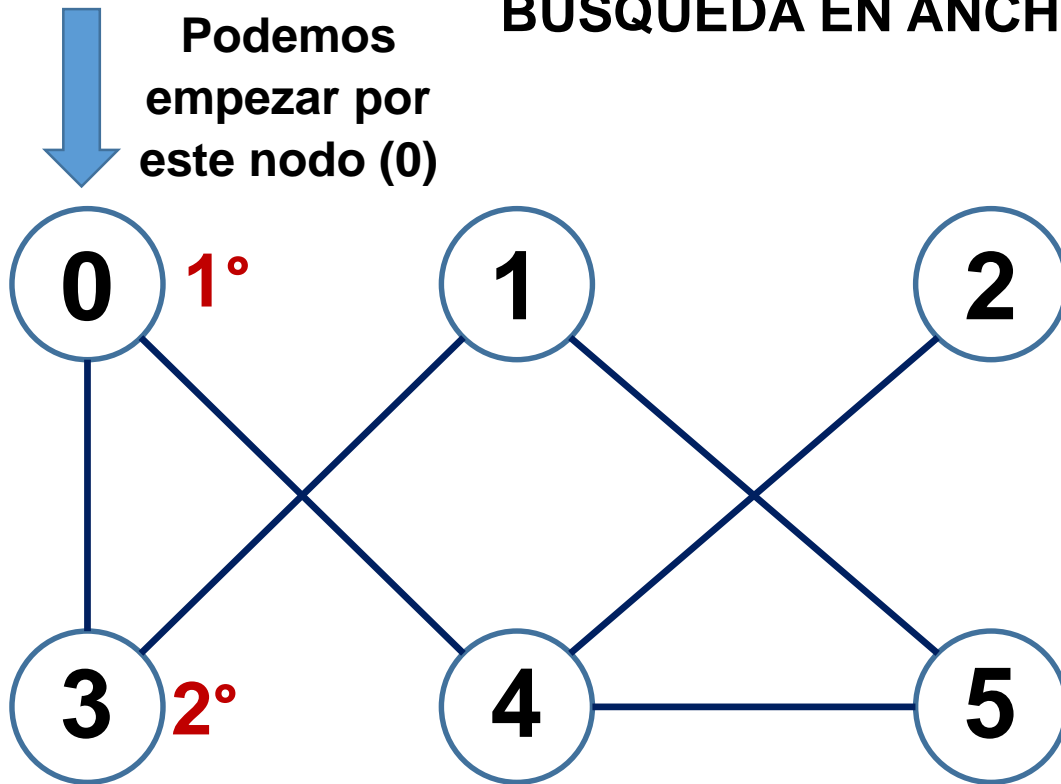
COLA
(QUEUE) → 4

Los adyacentes de 3 ya están marcados
Sacamos a 3 de la cola
(3 fue el 2º elemento en ser visitado)

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

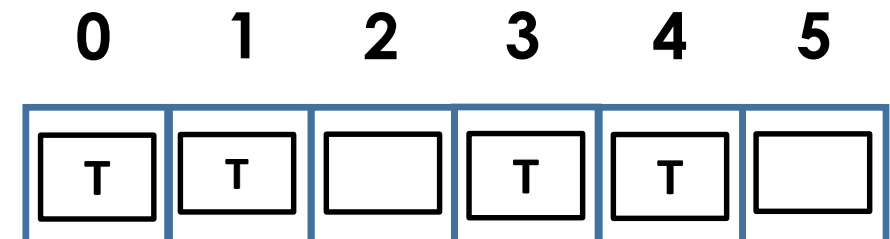


Nos apoyamos en una COLA para el recorrido por niveles

COLA (QUEUE) → 4
1

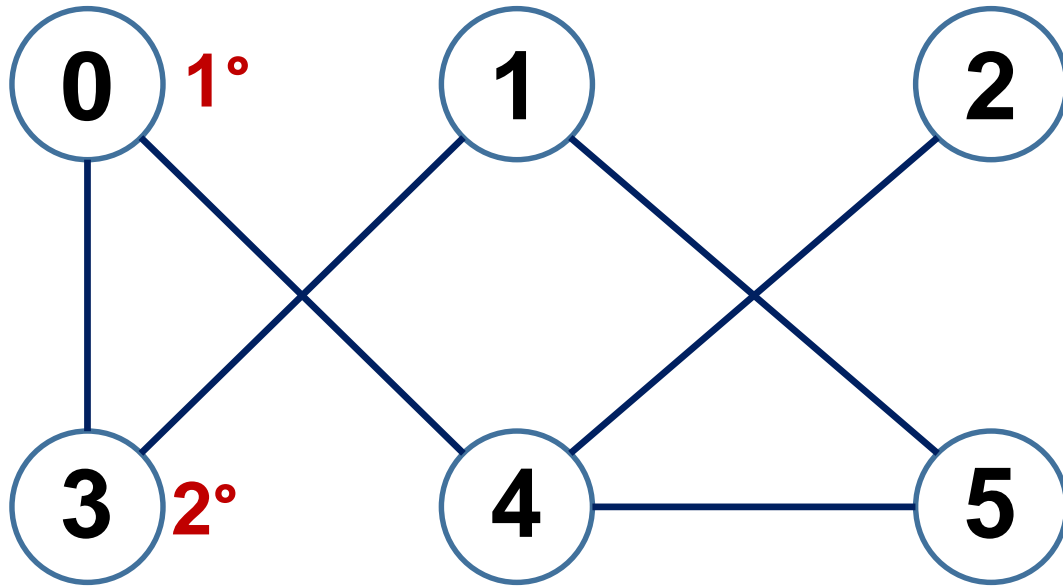
Insertamos el 1 a la cola

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

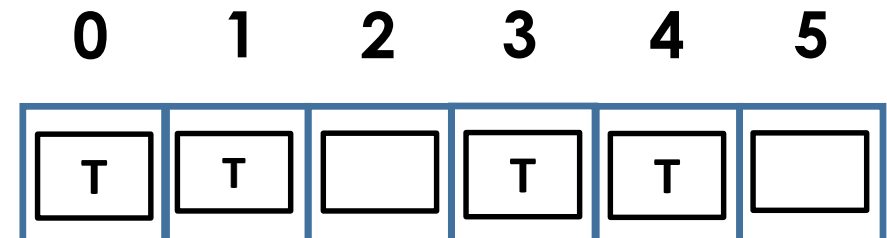


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) → 4
1

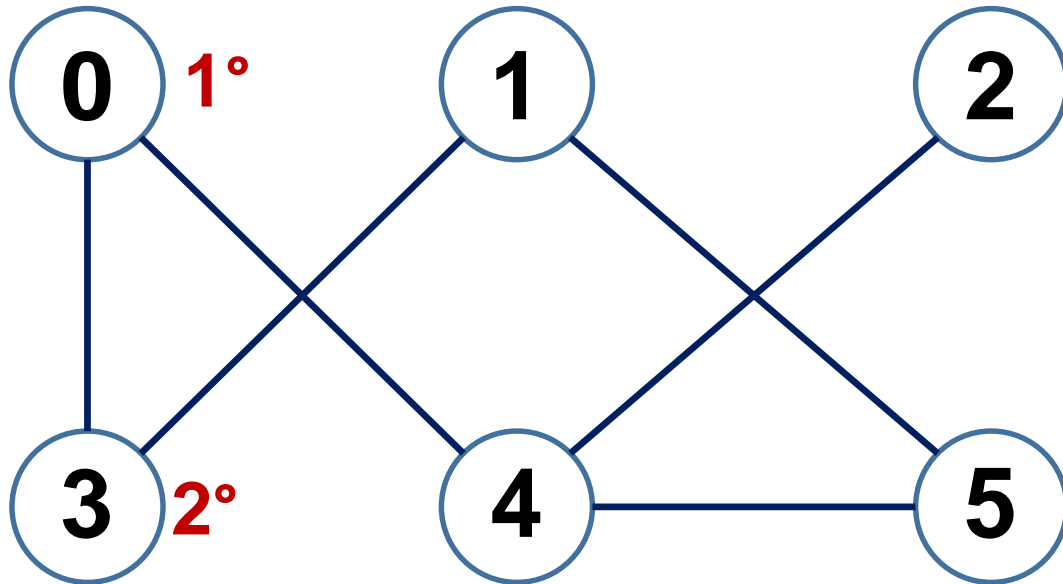
Analizamos el elemento 4.
Adyacentes: 0, 2, 5

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

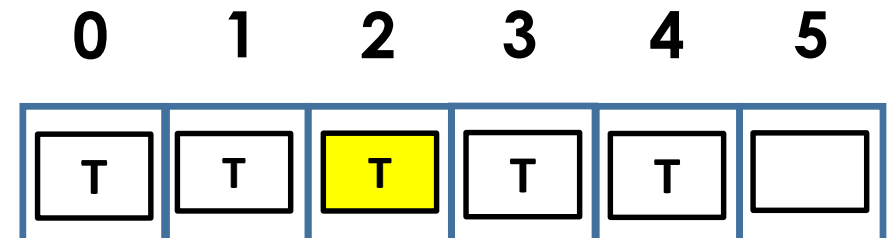


Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:

Nos apoyamos en una COLA
para el recorrido por niveles

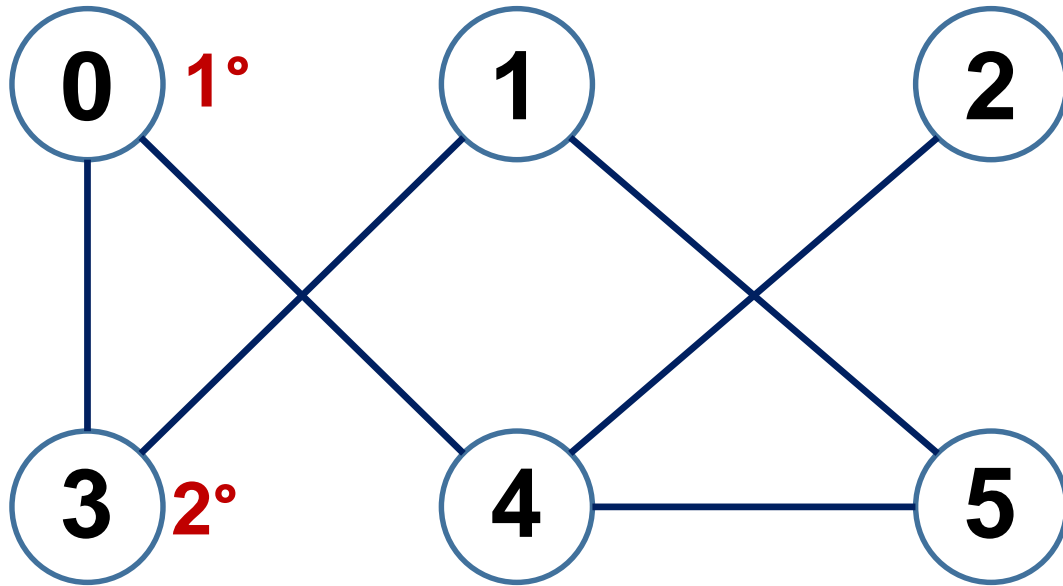
COLA
(QUEUE) → 4
1

0 ya fue visitado
Marcamos a 2 como visitado



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

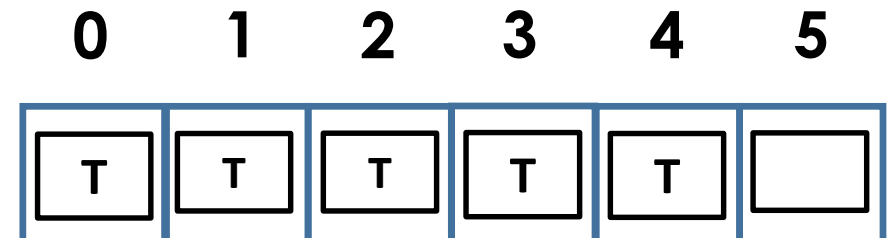


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) → 4
1
2

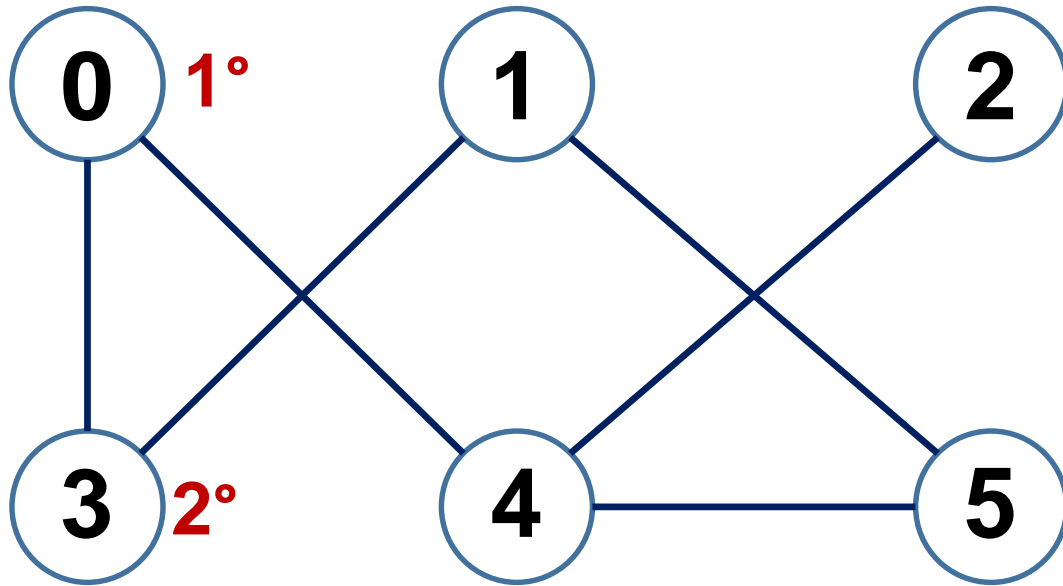
Insertamos el 2 a la cola

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

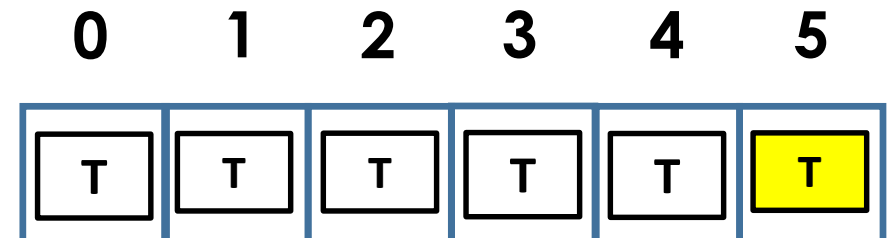


Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:

Nos apoyamos en una COLA
para el recorrido por niveles

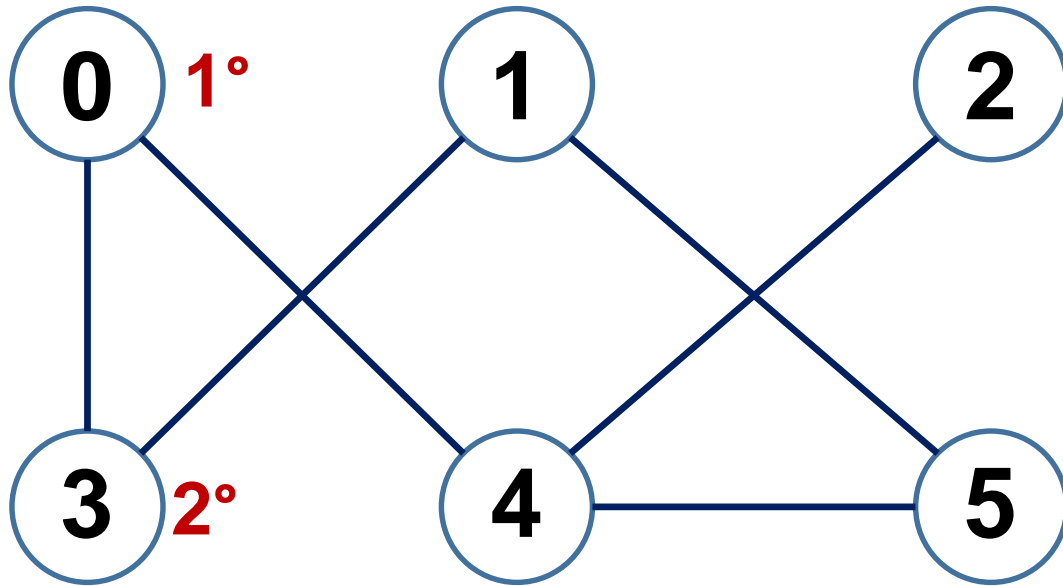
COLA
(QUEUE) →
4
1
2

Marcamos a 5 como visitado



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)



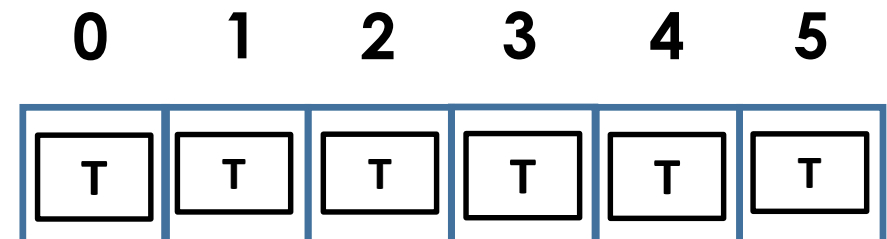
Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) →

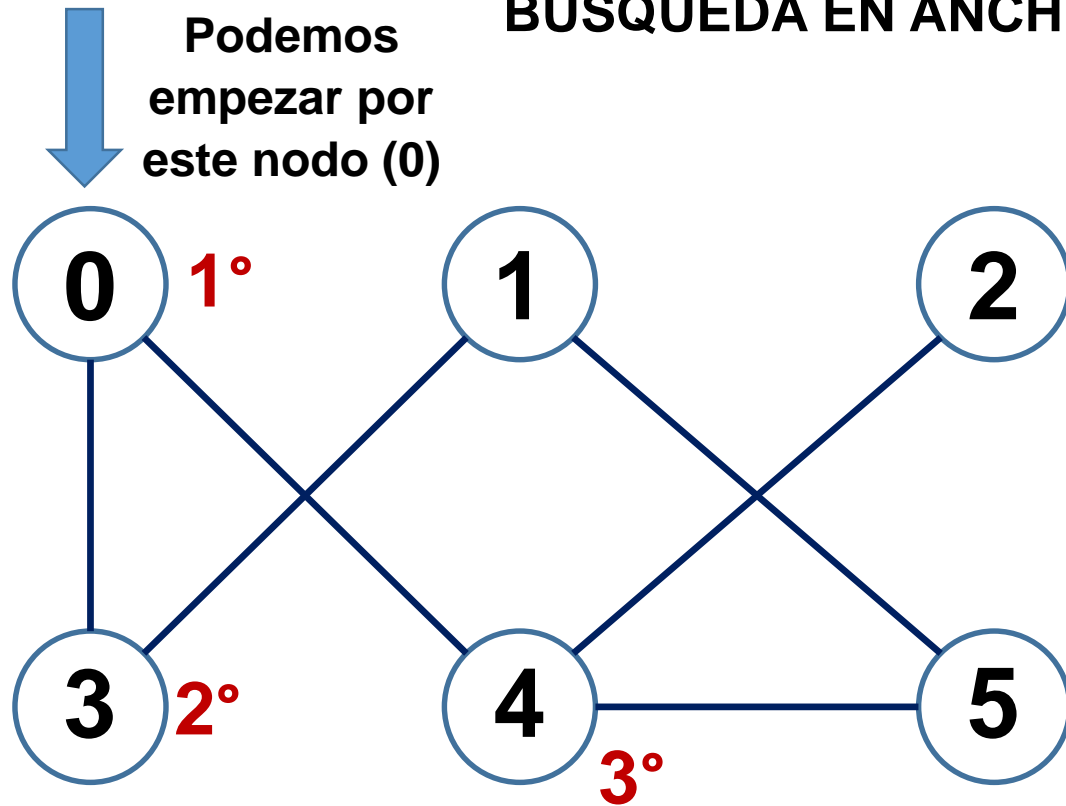
4
1
2
5

Insertamos el
5 a la cola

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

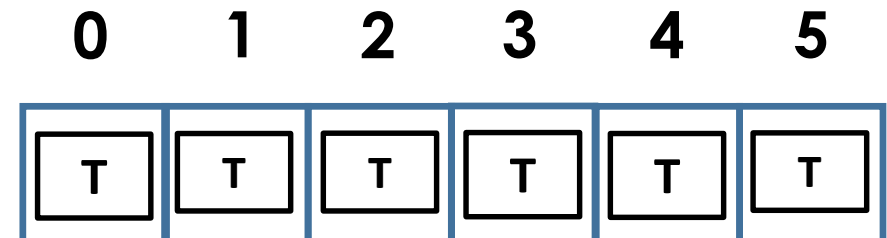


Nos apoyamos en una COLA para el recorrido por niveles

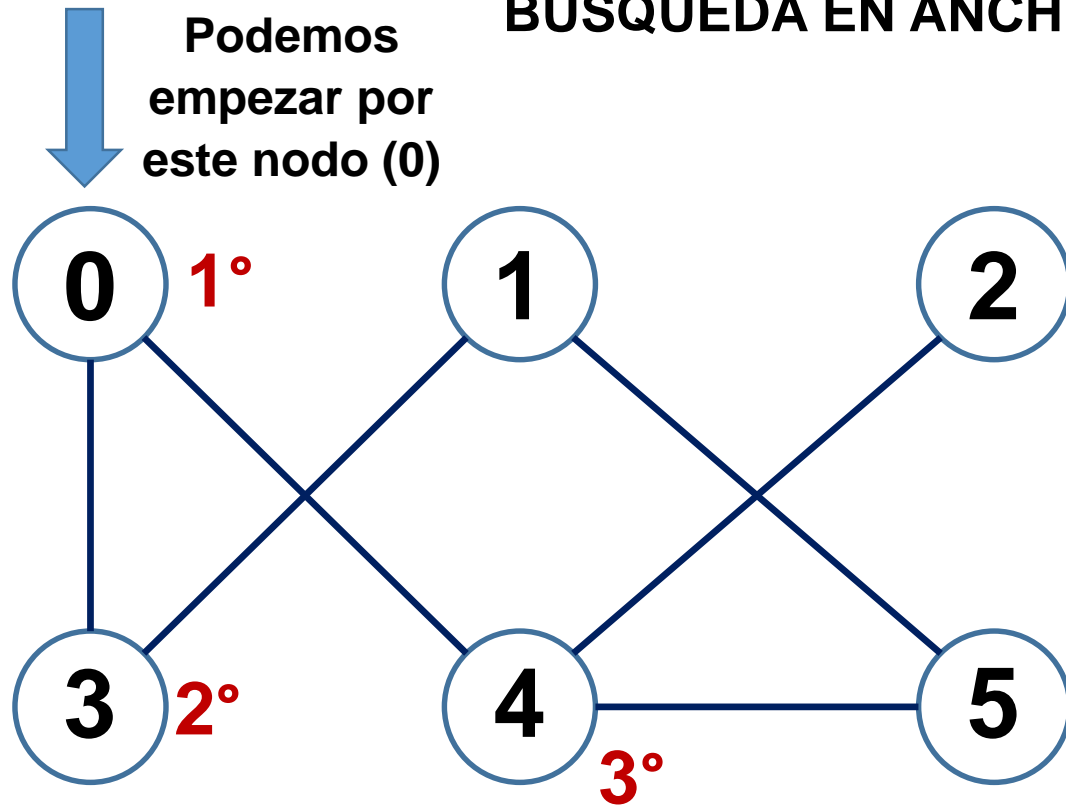
COLA (QUEUE) → 1
2
5

Los adyacentes de 4 ya están marcados
Sacamos a 4 de la cola
(4 fue el 3° elemento en ser visitado)

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

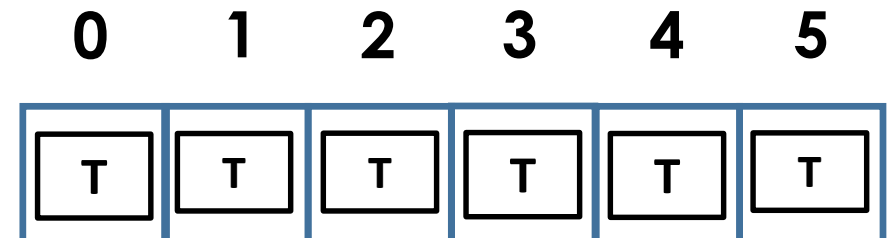


Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

Nos apoyamos en una COLA para el recorrido por niveles

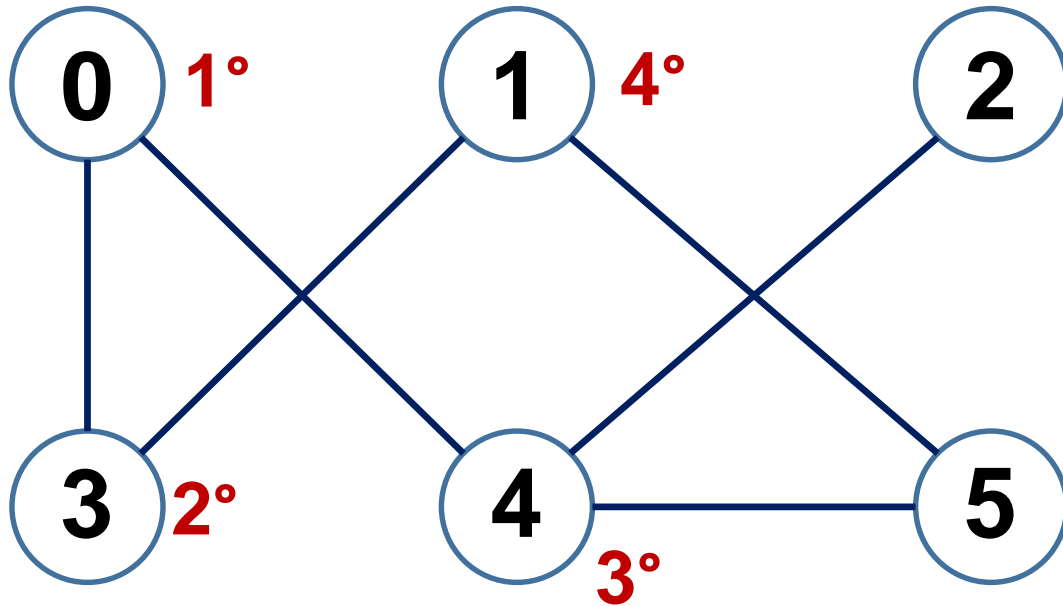
COLA (QUEUE) → 1
2
5

Analizamos el elemento 1.
Adyacentes: 3 , 5



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

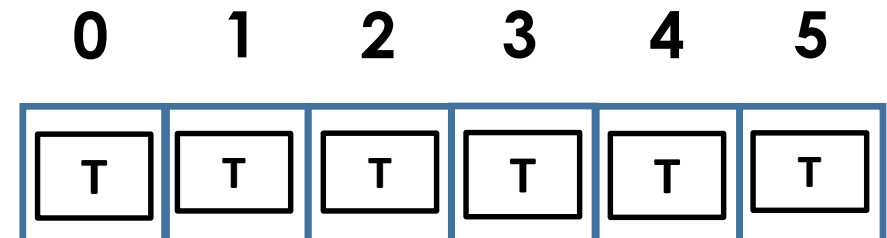


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) → 2
5

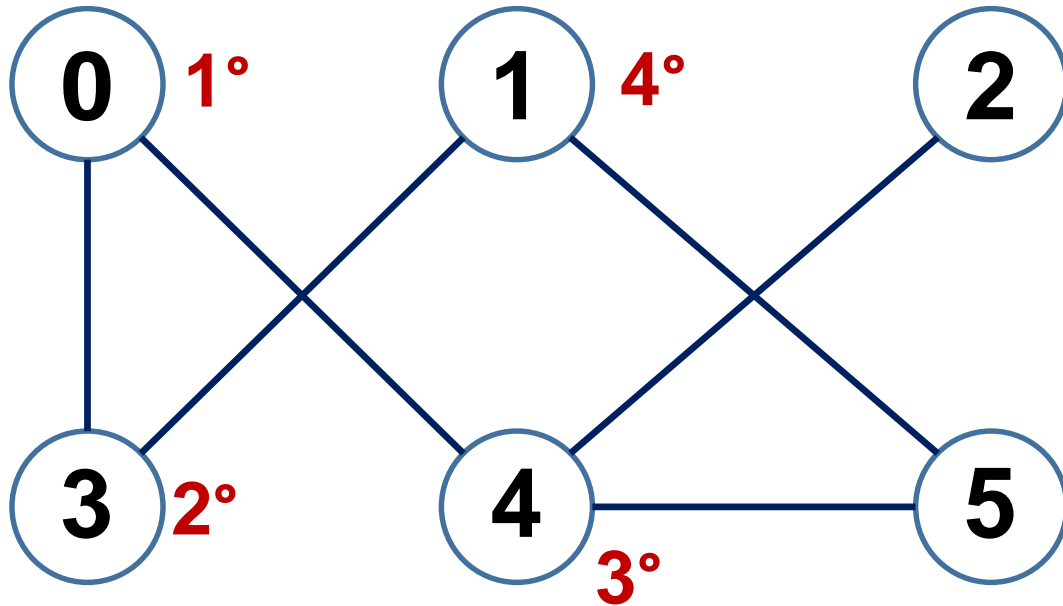
Los adyacentes de 1
ya están marcados
Sacamos a 1 de la cola
(1 fue el 4° elemento
en ser visitado)

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

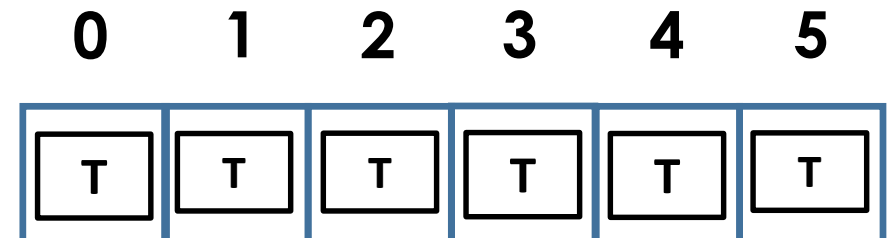


Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:

Nos apoyamos en una COLA
para el recorrido por niveles

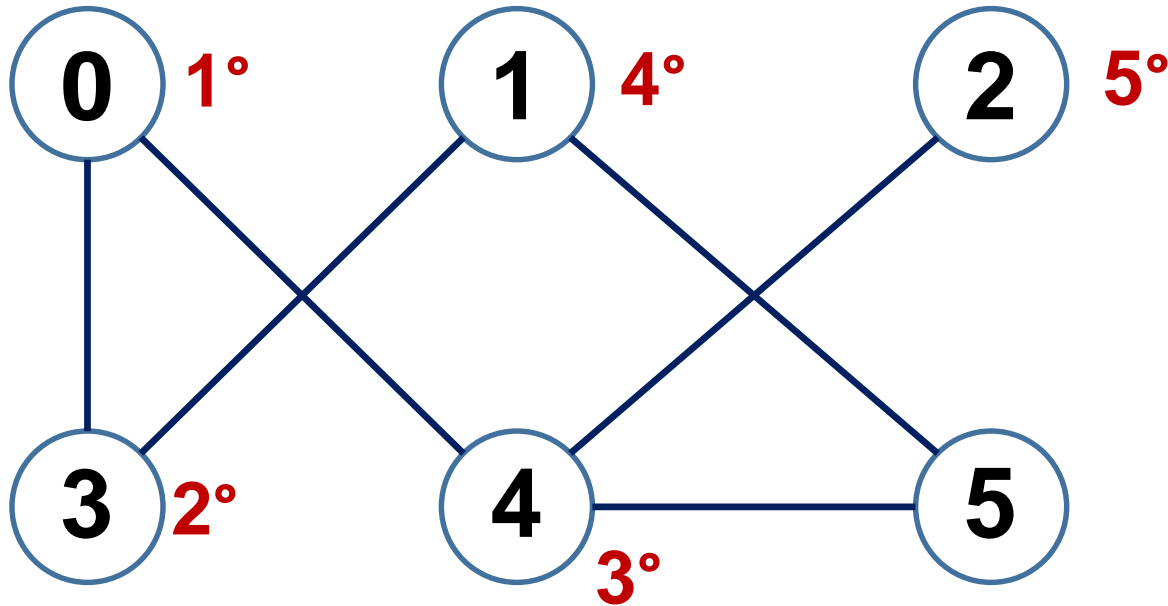
COLA
(QUEUE) → 2
5

Analizamos el elemento 2.
Adyacentes: 4



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

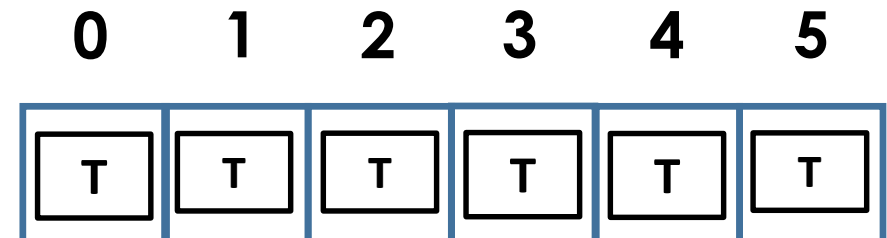


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) → 5

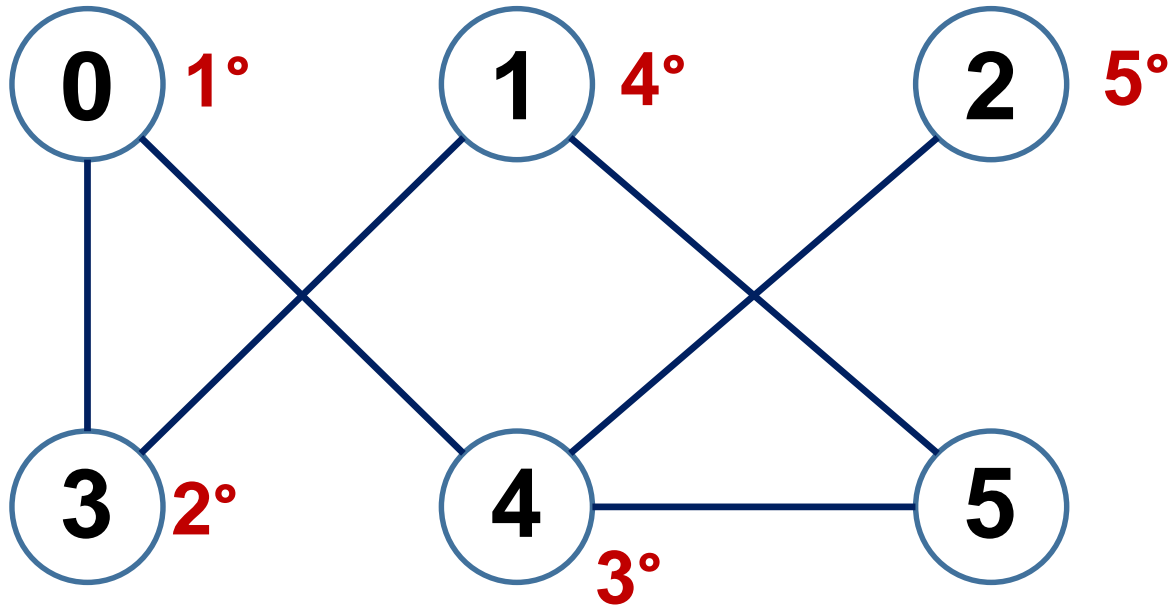
El adyacente de 2
ya está marcado
Sacamos a 2 de la cola
(2 fue el 5° elemento
en ser visitado)

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

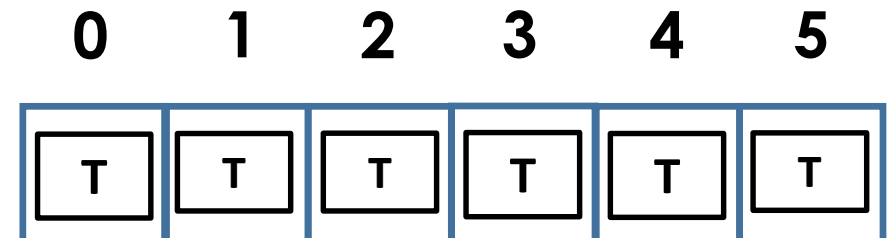


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) → 5

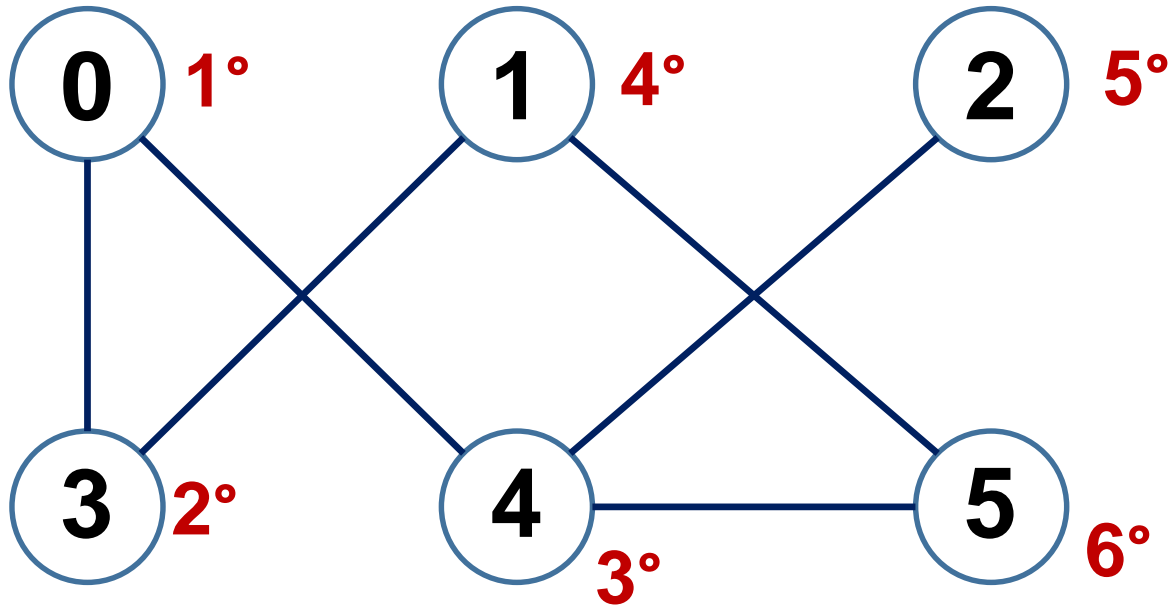
Analizamos el elemento 5.
Adyacentes: 1 , 4

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

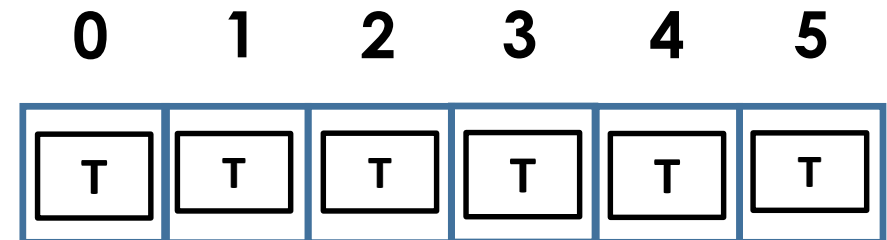


Nos apoyamos en una COLA
para el recorrido por niveles

COLA
(QUEUE) →

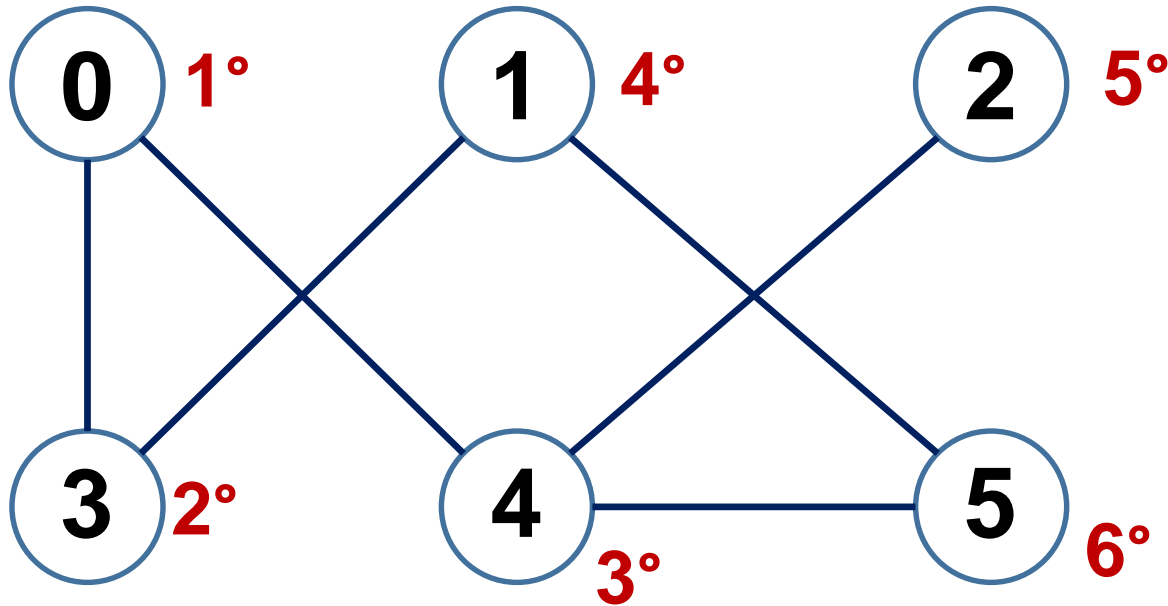
Los adyacentes de 5
ya están marcados
Sacamos a 5 de la cola
(5 fue el 6° elemento
en ser visitado)

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Podemos
empezar por
este nodo (0)

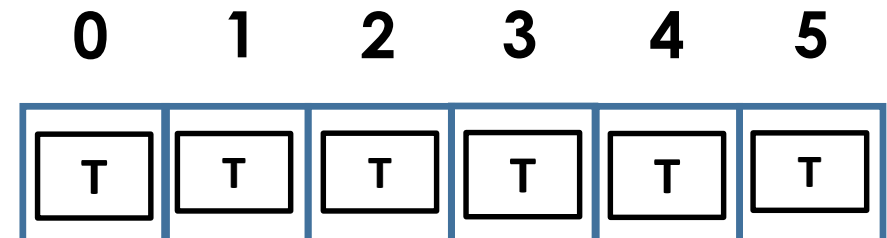


Nos apoyamos en una COLA
para el recorrido por niveles

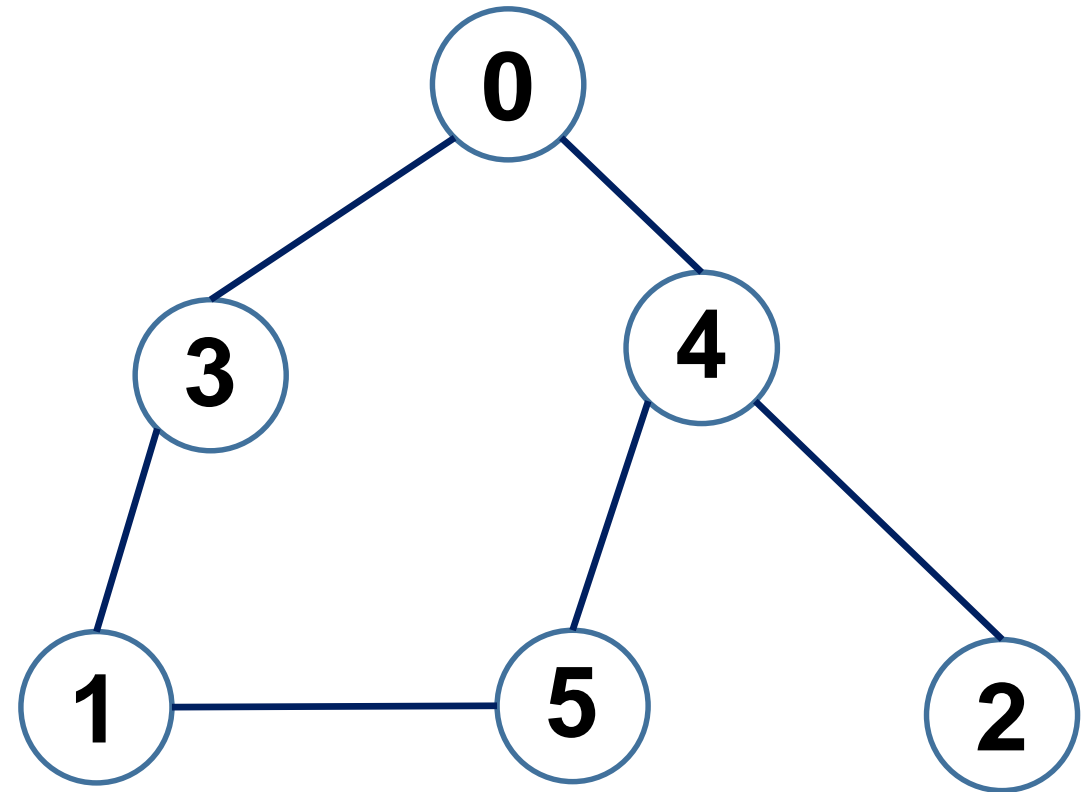
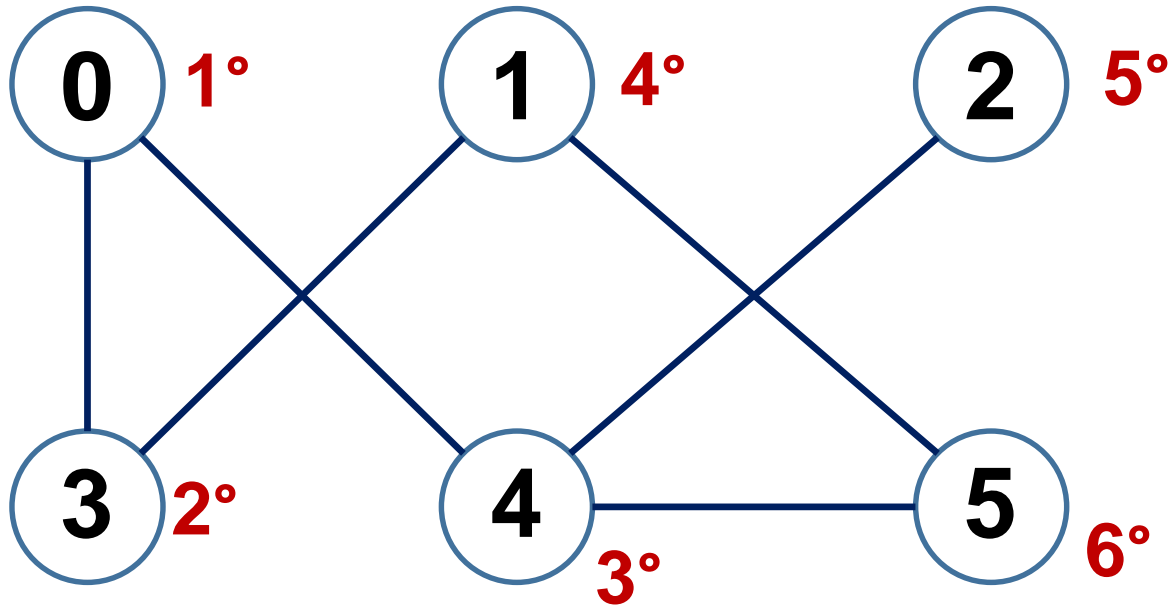
COLA
(QUEUE) →

**COLA QUEDÓ VACÍA.
RECORRIDO TERMINADO.**

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)



RECORRIDO DEL GRAFO EN ANCHURA BFS

0 – 3 – 4 – 1 – 2 – 5

BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

COMPLEJIDAD COMPUTACIONAL EN FUNCIÓN DEL TIEMPO

Si el grafo está representado como una

lista de adyacencia

la complejidad es $O(V + E)$

Si el grafo está representado como una

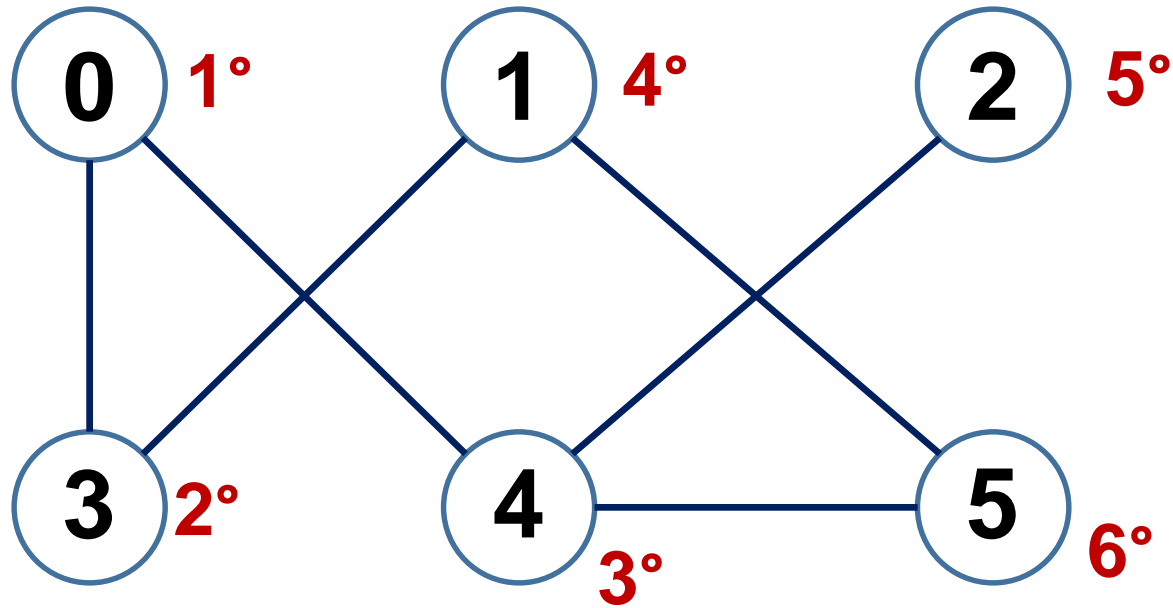
matriz de adyacencia

la complejidad es $O(V * V)$

BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Una de sus principales aplicaciones es hallar el camino más corto entre 2 vértices

Para el grafo de nuestro ejemplo:

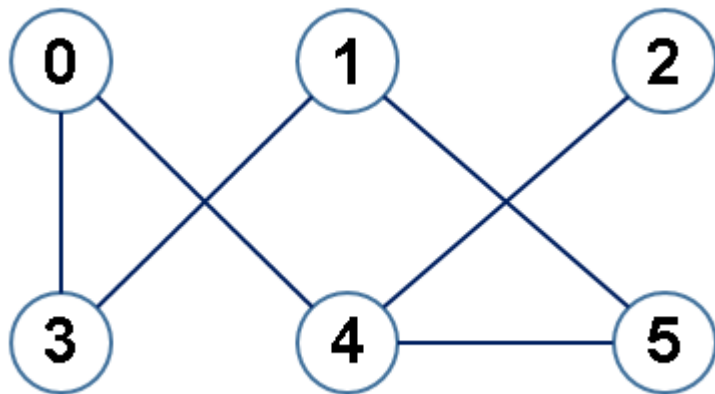


¿Cómo hallar el camino más corto entre los vértices 0 y 5?

BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

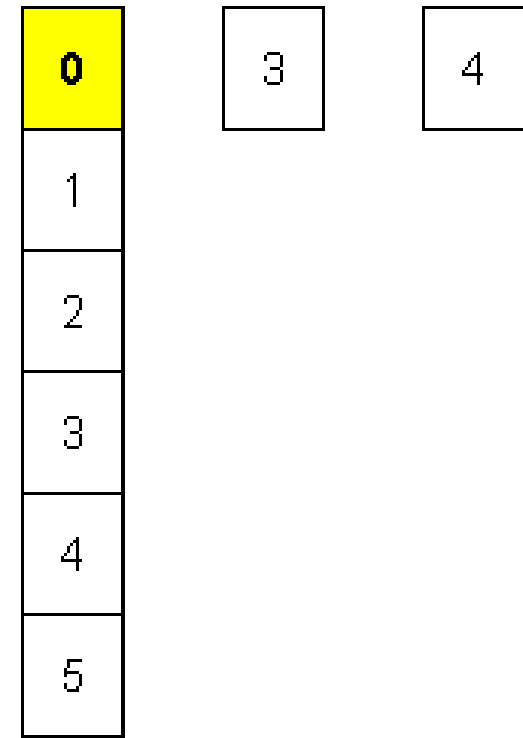
Una de sus principales aplicaciones es hallar el camino más corto entre 2 vértices

La respuesta es sencilla: al hacer el recorrido iremos construyendo la lista de adyacencia, y cuando aparezca el vértice de destino por primera vez, lo habremos hallado



COLA
(QUEUE) →

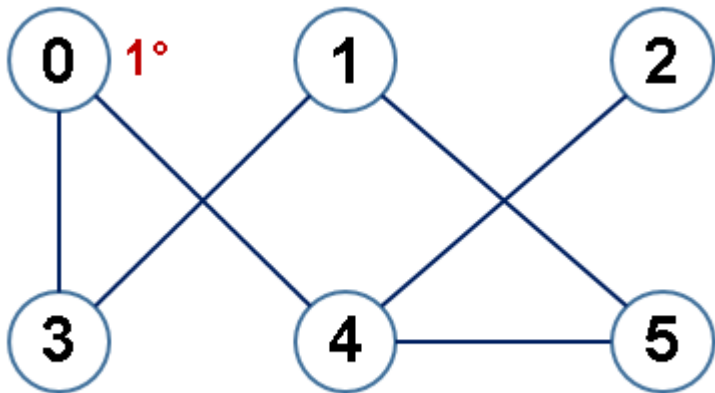
Adyacentes de 0: son 3 y 4.



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

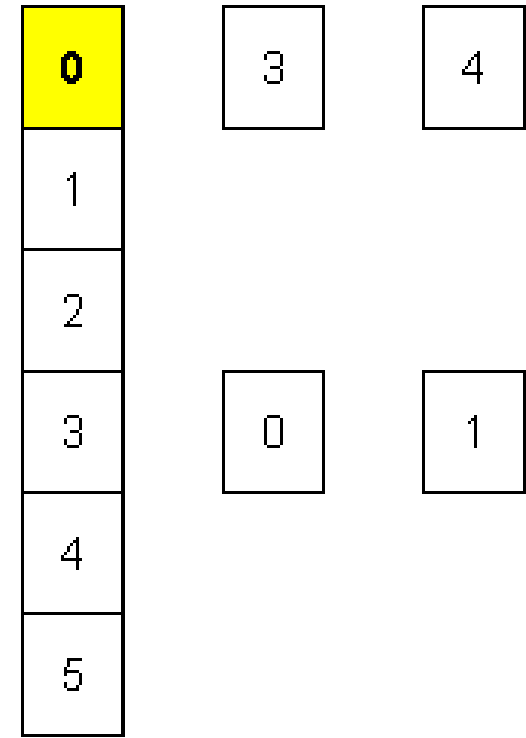
Una de sus principales aplicaciones es hallar el camino más corto entre 2 vértices

La respuesta es sencilla: al hacer el recorrido iremos construyendo la lista de adyacencia, y cuando aparezca el vértice de destino por primera vez, lo habremos hallado



COLA
(QUEUE) → 3
4

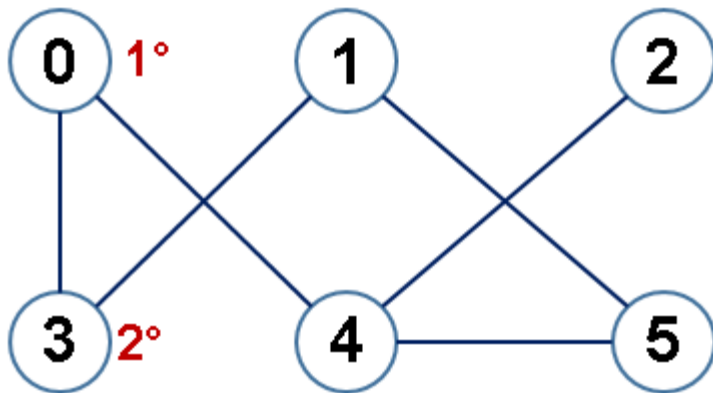
Cola NO está vacía.
Expandimos a los adyacentes de 3
Que son 0 y 1



BÚSQUEDA EN ANCHURA BFS (Breadth First Search)

Una de sus principales aplicaciones es hallar el camino más corto entre 2 vértices

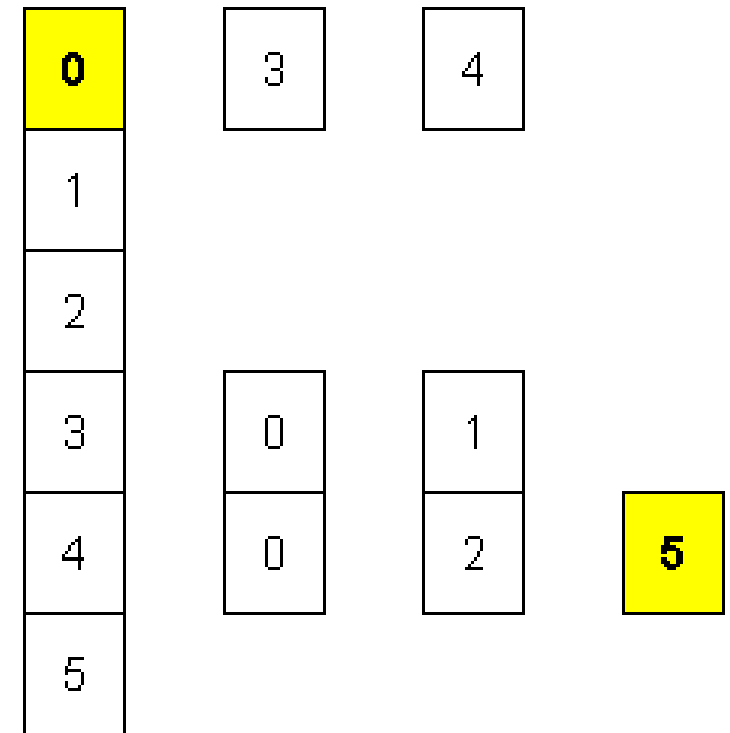
La respuesta es sencilla: al hacer el recorrido iremos construyendo la lista de adyacencia, y cuando aparezca el vértice de destino por primera vez, lo habremos hallado



COLA
(QUEUE) →

4
1

Analizamos el elemento 4.
Adyacentes: 0, 2, 5



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

Se comienza con un nodo u

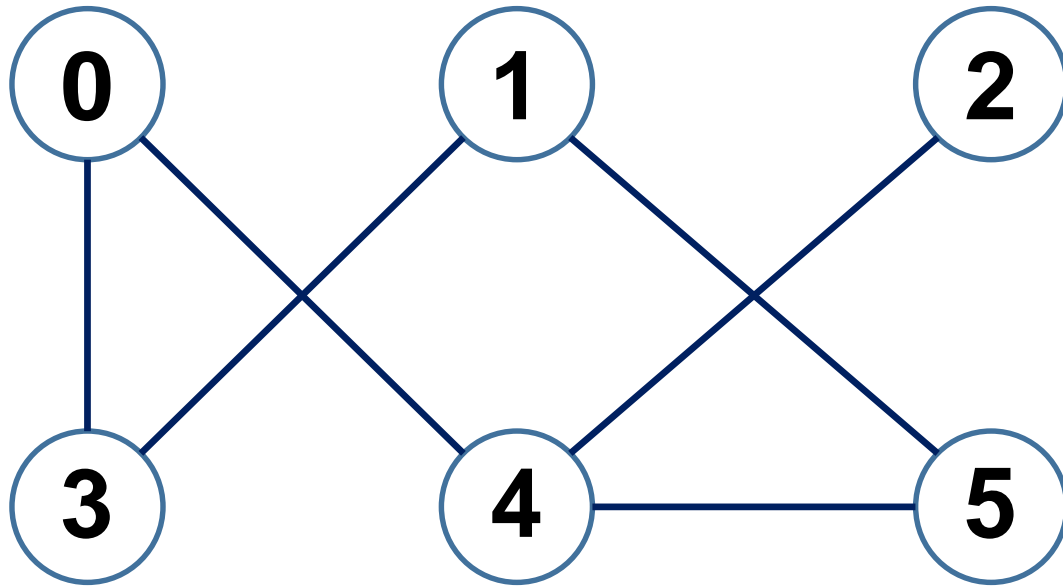
Por cada nodo se expande de manera recursiva
en un camino concreto

Cuando no es posible visitar más nodos por ese camino,
se devuelve

Se repite el proceso con cada uno de los nodos hermanos
del nodo ya procesado

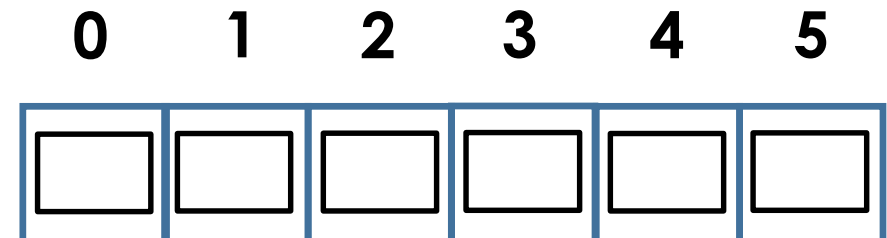
BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

Nos apoyamos en una PILA para el recorrido en profundidad



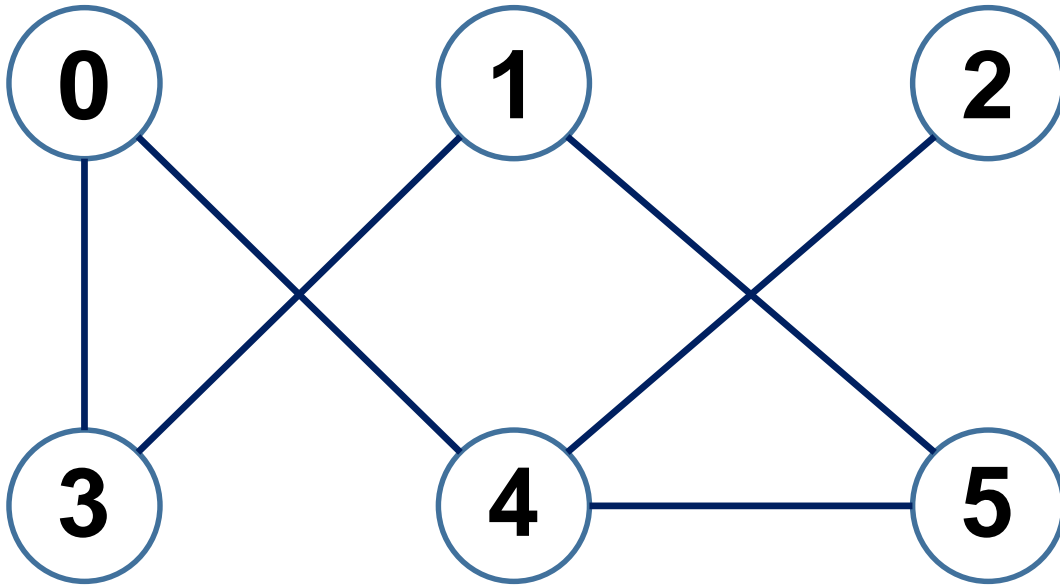
Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

PILA (STACK) →



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

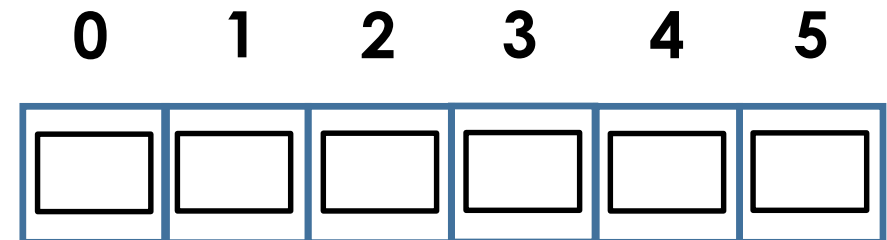
Podemos
empezar por
este nodo (0)



Nos apoyamos en una PILA
para el recorrido en profundidad

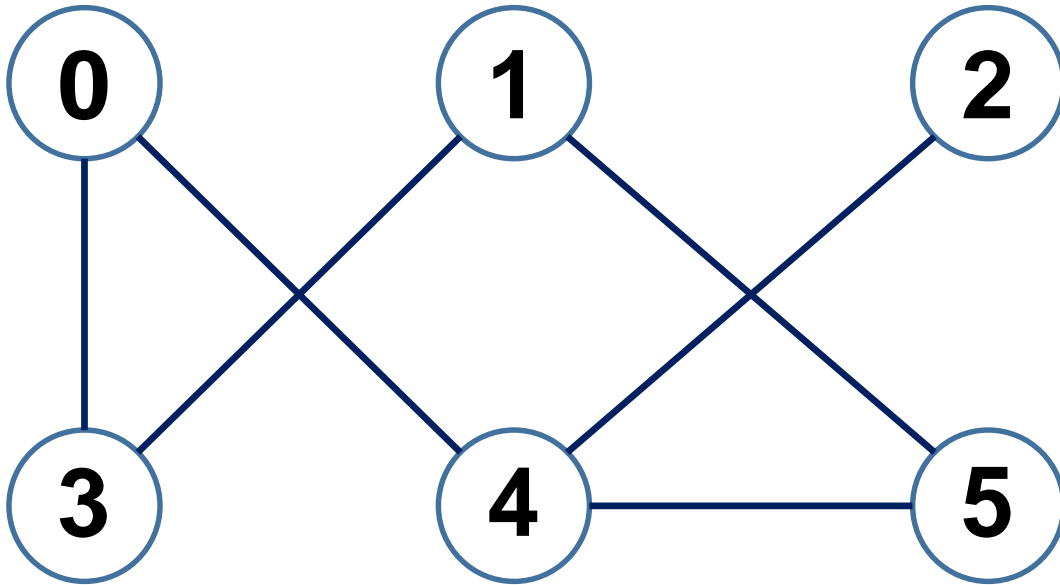
PILA
(STACK) →

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

Podemos
empezar por
este nodo (0)

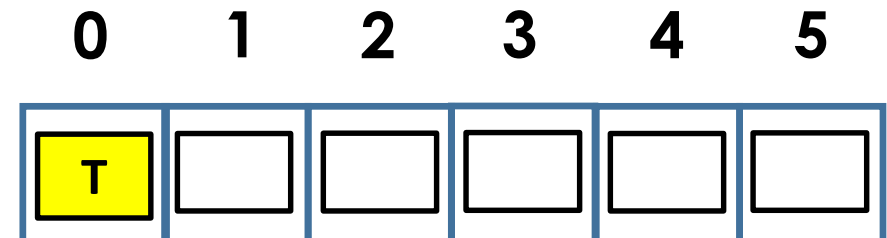


Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

Nos apoyamos en una PILA para el recorrido en profundidad

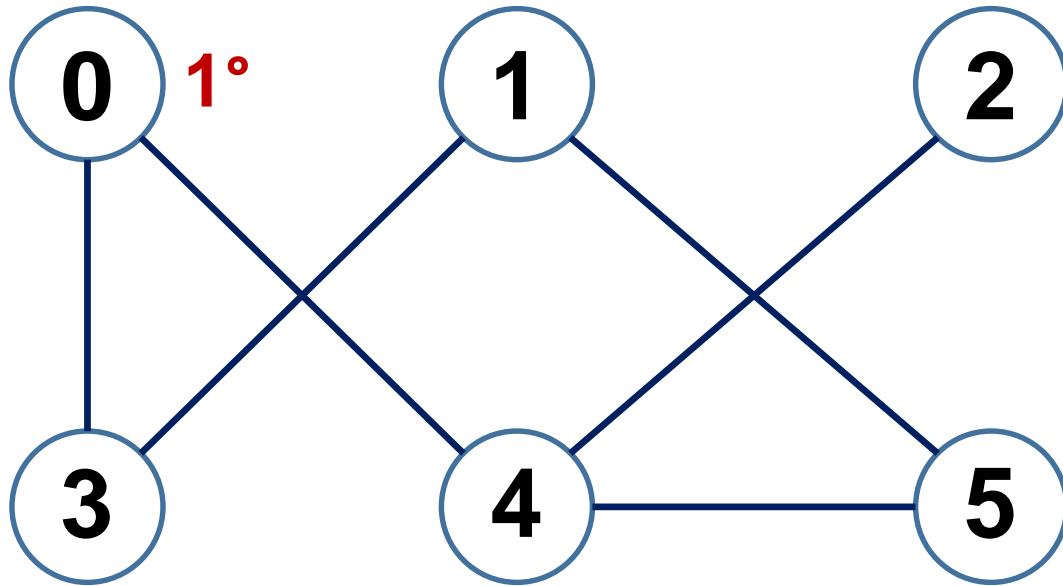
PILA (STACK) →

Lo marcamos como visitado



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

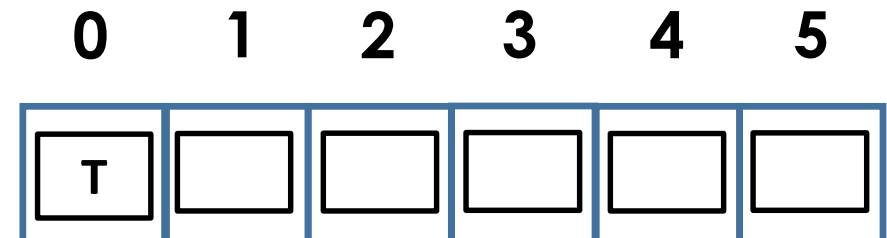
Podemos
empezar por
este nodo (0)



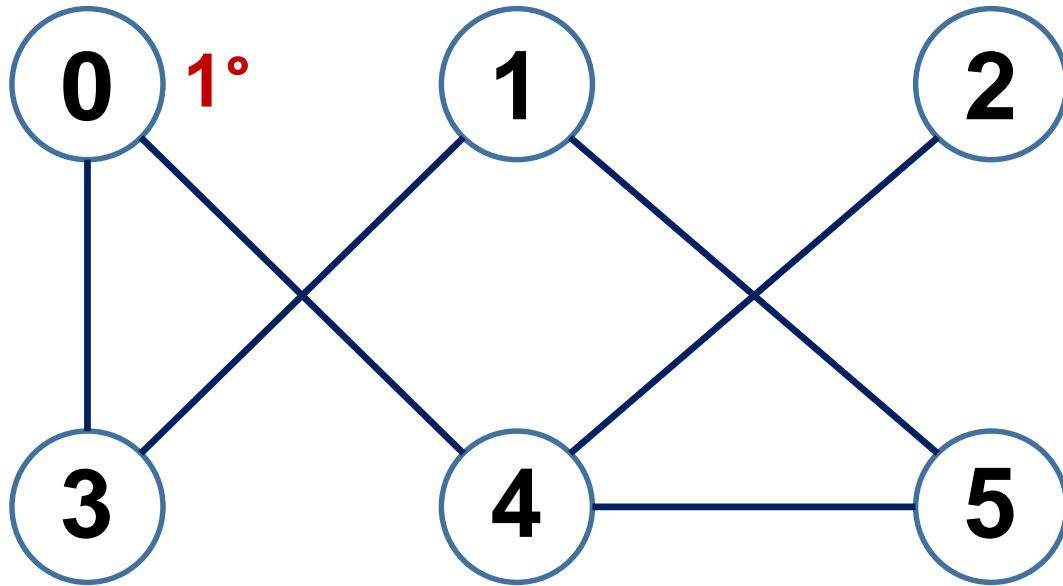
Nos apoyamos en una PILA
para el recorrido en profundidad

PILA (STACK) → **0** Lo insertamos en PILA

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

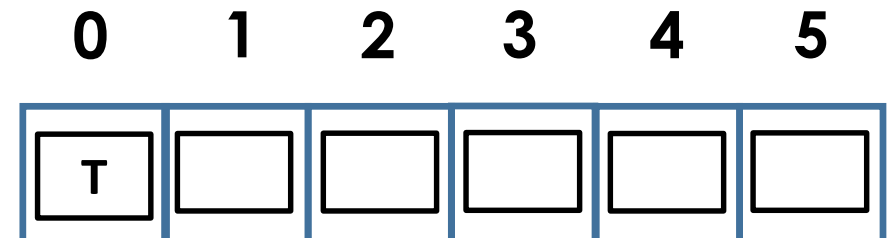


Nos apoyamos en una PILA para el recorrido en profundidad

PROCEDIMIENTO:
Mientras la pila NO esté vacía
Analizamos adyacentes

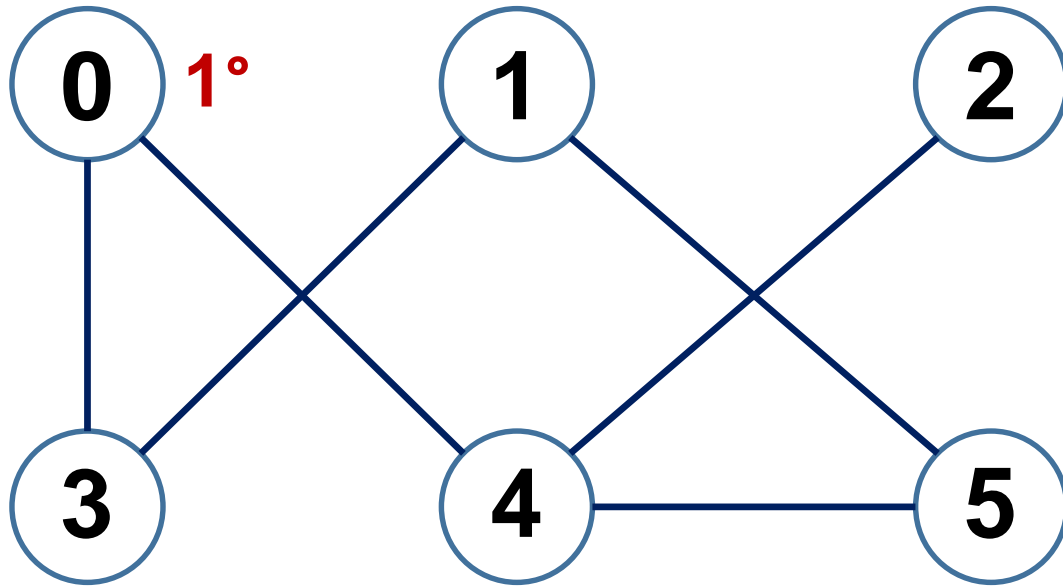
PILA (STACK) → 0

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

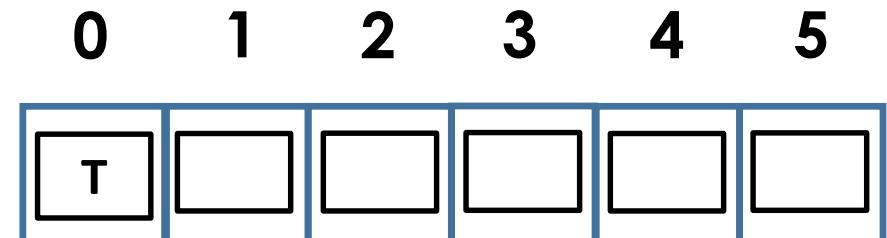
Nos apoyamos en una PILA para el recorrido en profundidad



Adyacentes de 0: son 3 y 4.
Analizamos 3.

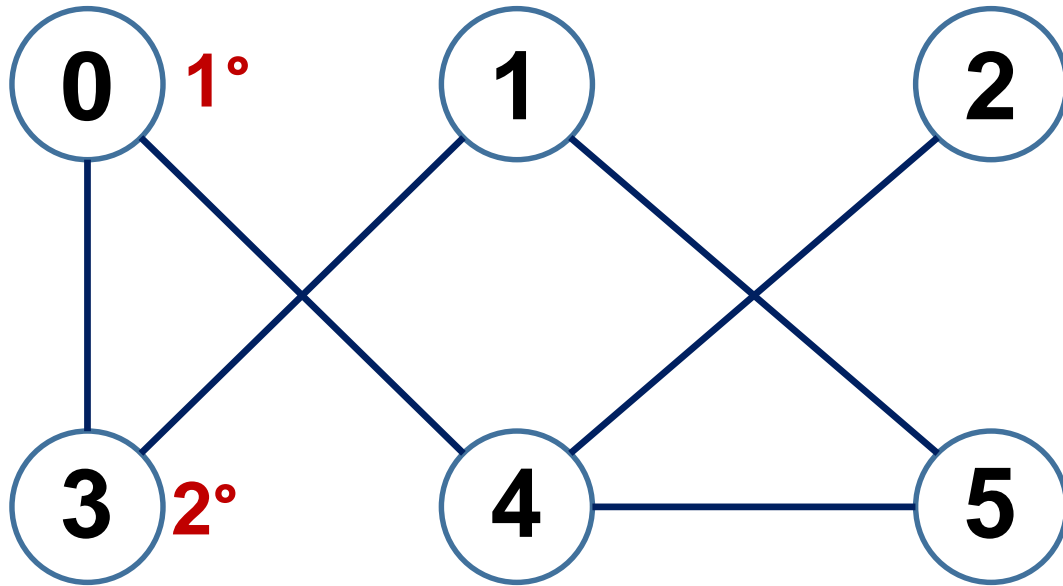
PILA (STACK) → 0

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

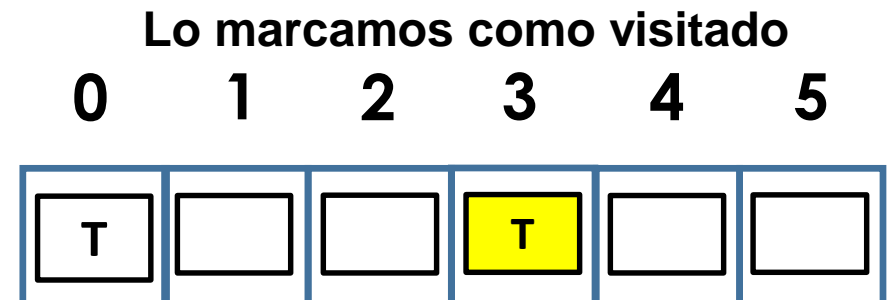
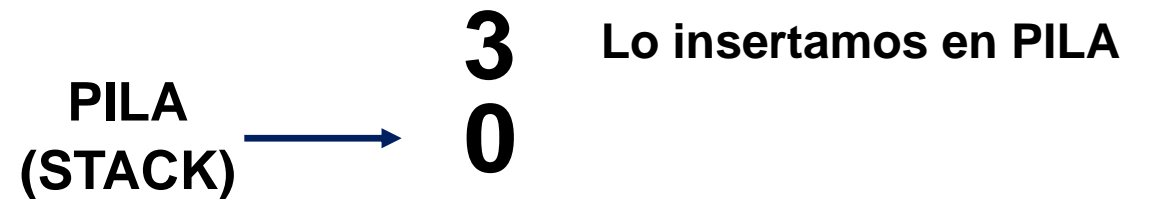


BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

Nos apoyamos en una PILA para el recorrido en profundidad

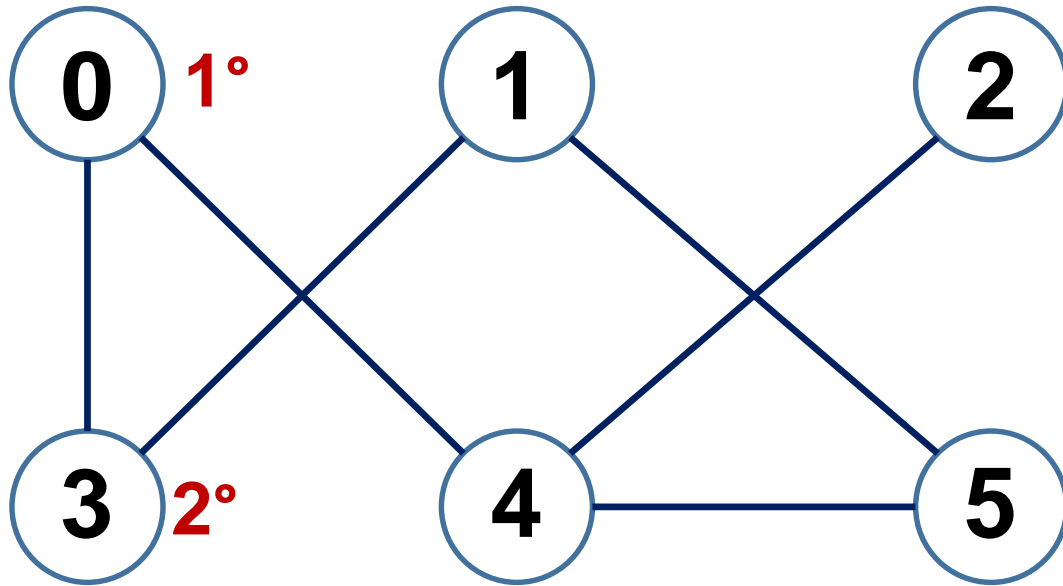


Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

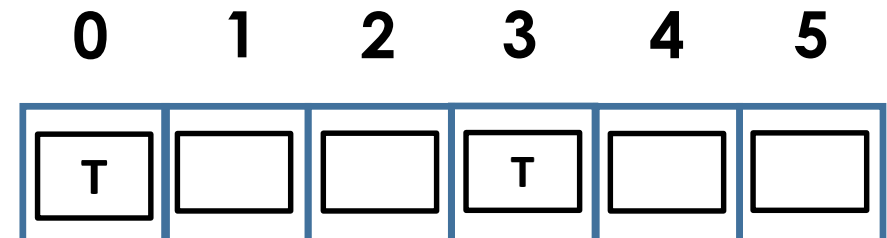
Nos apoyamos en una PILA para el recorrido en profundidad



Adyacentes de 3: 0 , 1
0 ya fue visitado.
Analizamos 1.

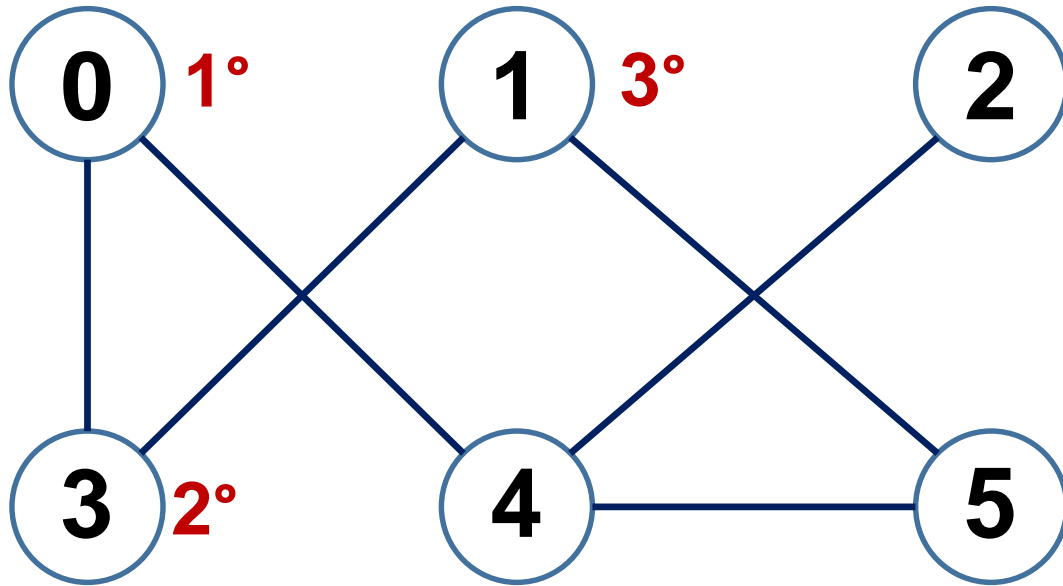


Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

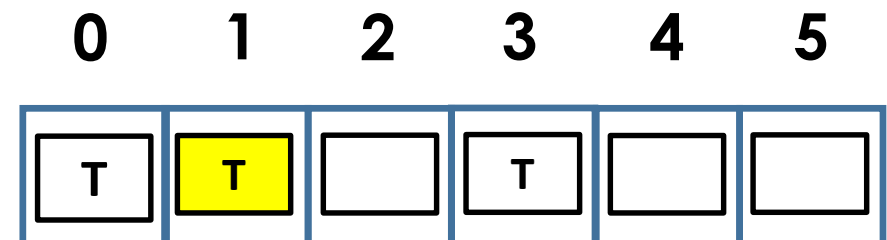
Nos apoyamos en una PILA para el recorrido en profundidad



Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

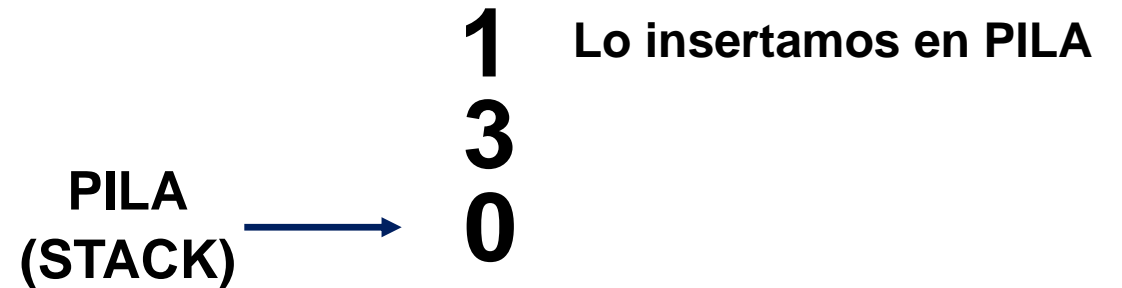
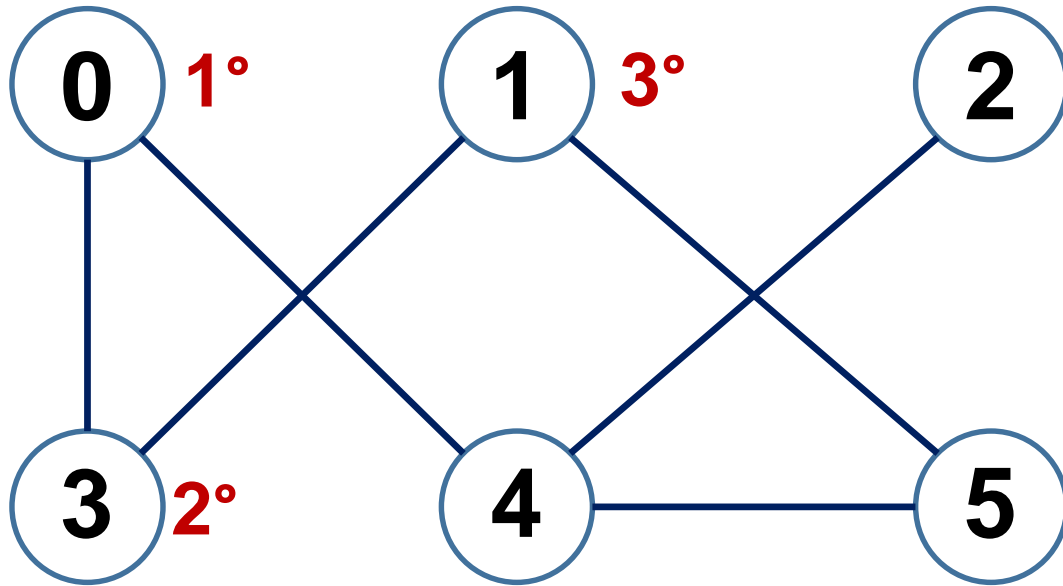


Lo marcamos como visitado

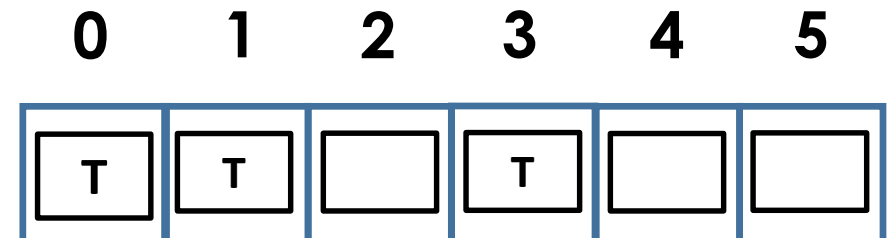


BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

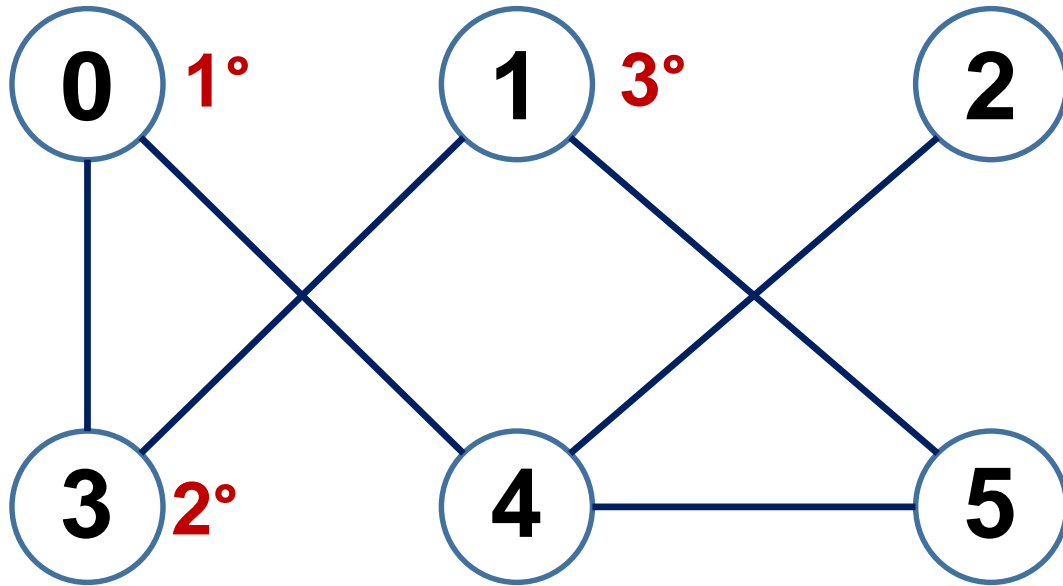
Nos apoyamos en una PILA para el recorrido en profundidad



Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



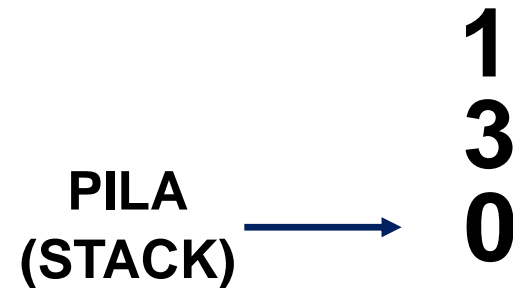
BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



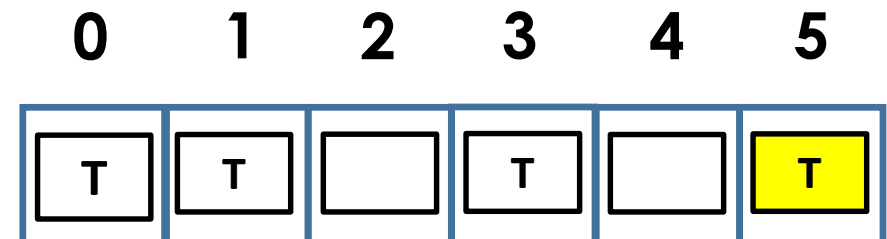
Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

Nos apoyamos en una PILA para el recorrido en profundidad

Adyacentes de 1: 3, 5
3 ya fue visitado.
Analizamos 5.

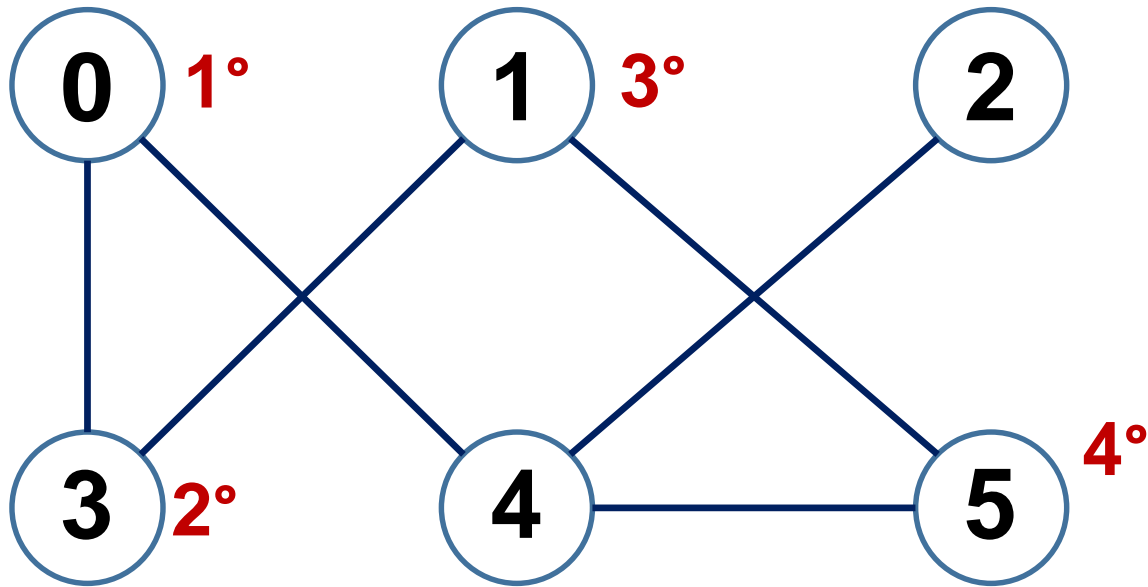


Lo marcamos como visitado



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

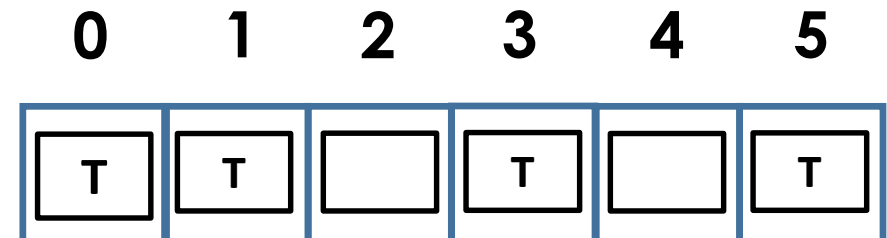
Nos apoyamos en una PILA para el recorrido en profundidad



PILA (STACK) →

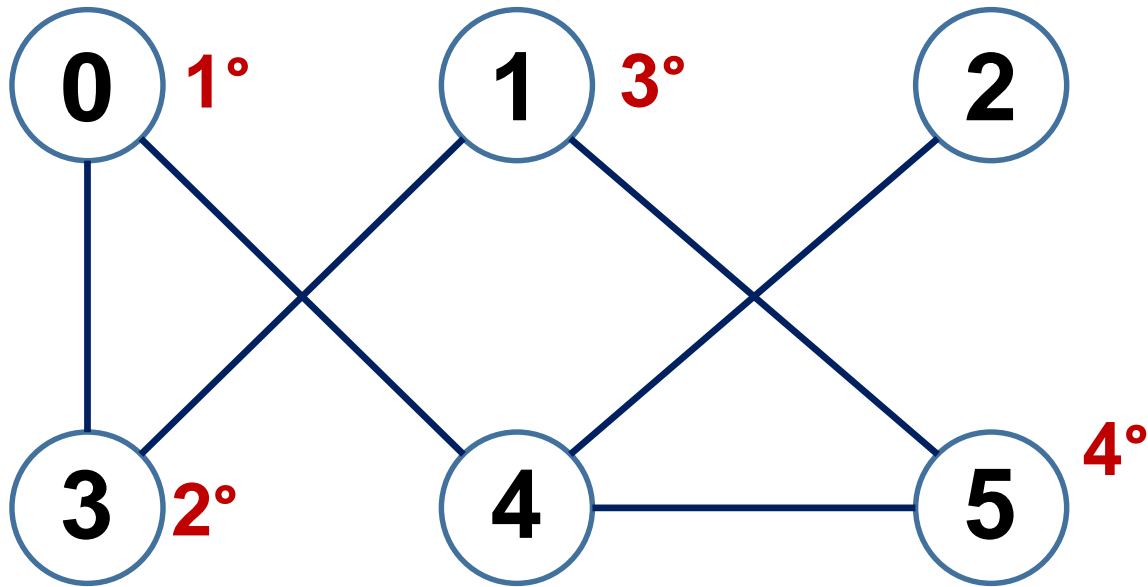
5 Lo insertamos en PILA
1
3
0

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

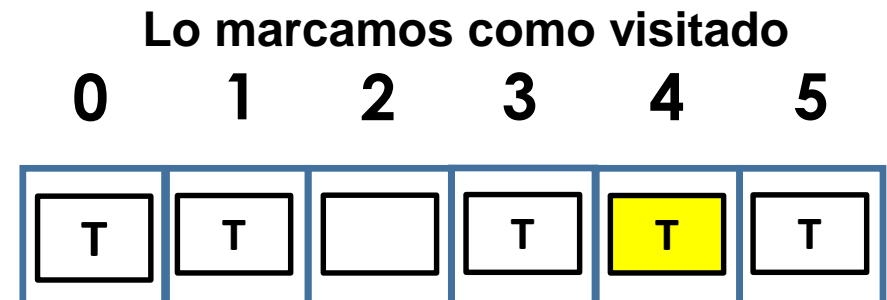
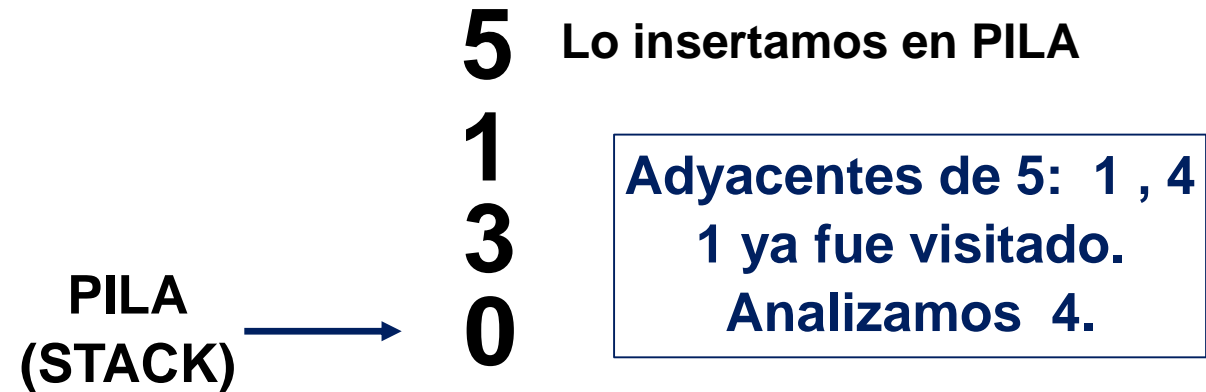


BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

Nos apoyamos en una PILA para el recorrido en profundidad

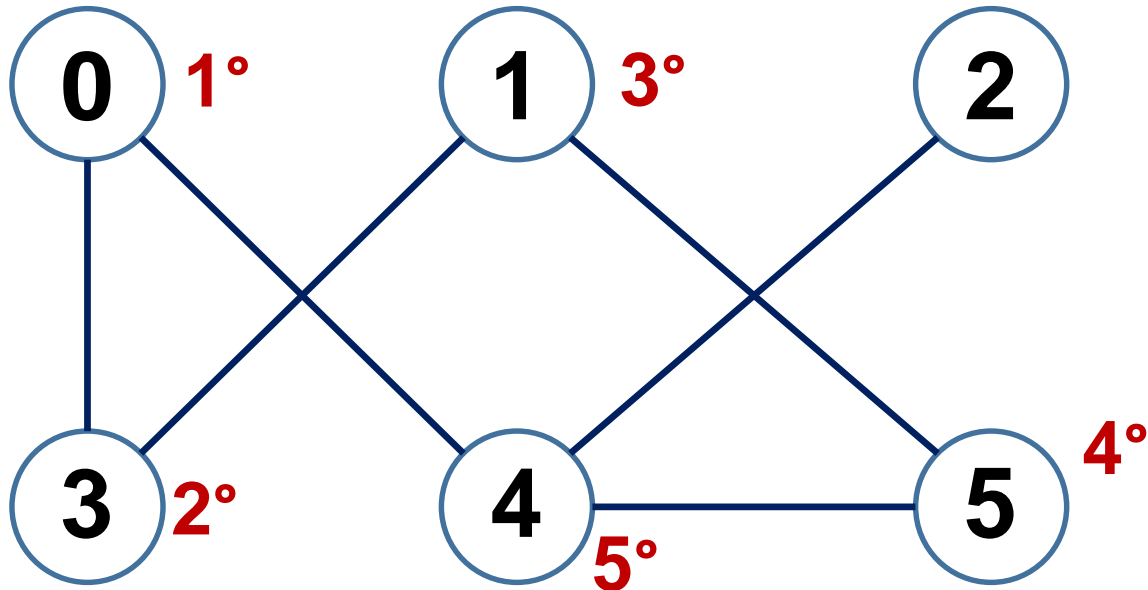


Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

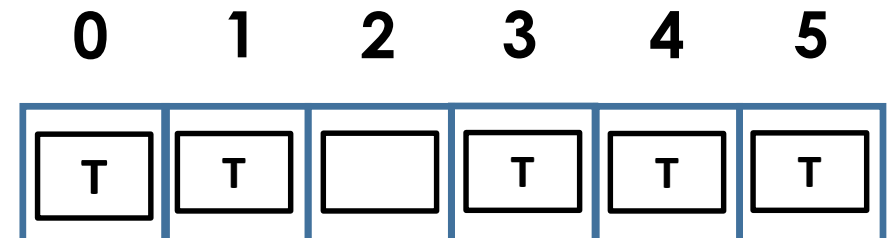
Nos apoyamos en una PILA para el recorrido en profundidad



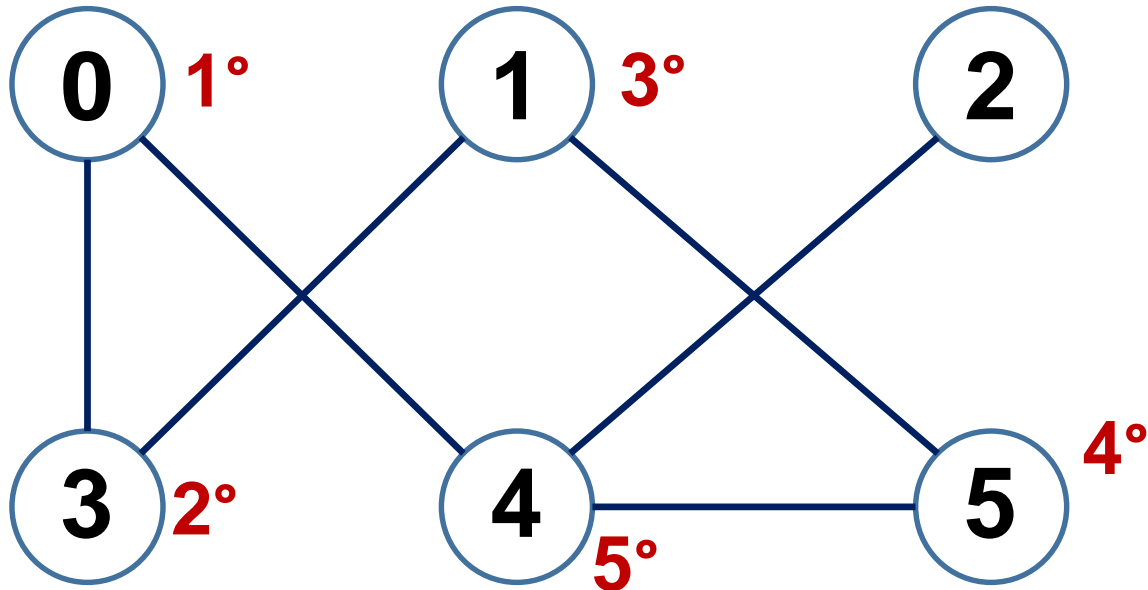
PILA (STACK) →

4 Lo insertamos en PILA
5
1
3
0

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

Nos apoyamos en una PILA para el recorrido en profundidad

PILA (STACK) →

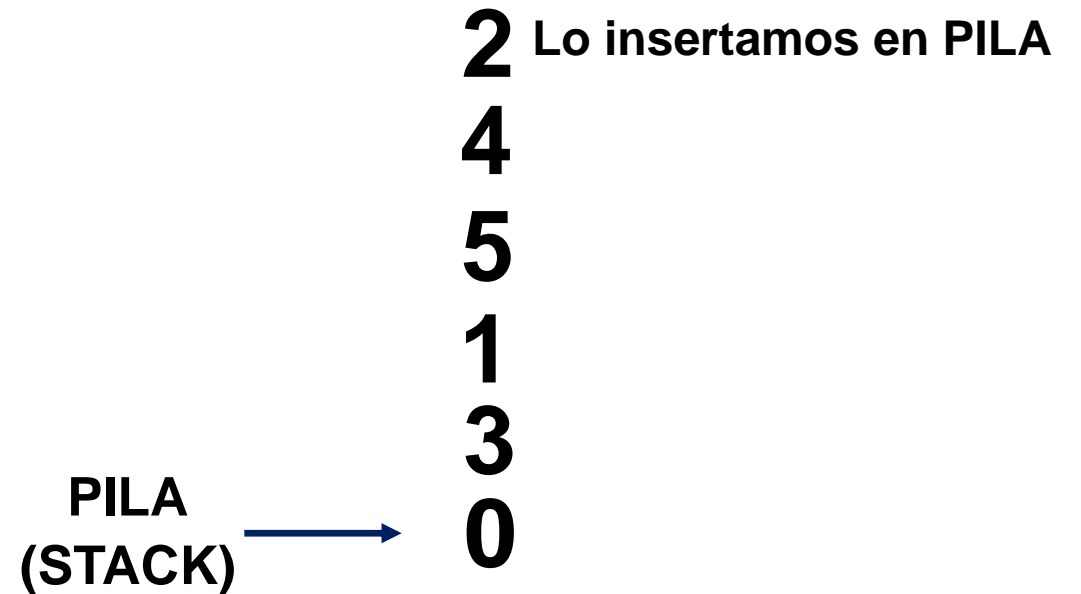
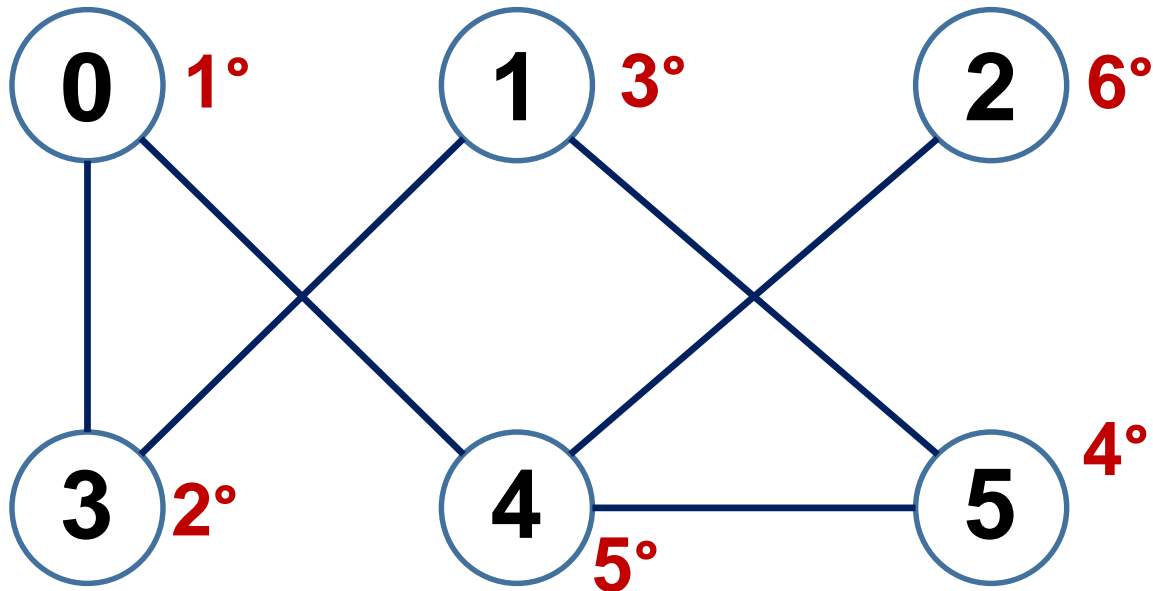
4
5
1
3
0

Adyacentes de 4: 0, 2, 5
0 ya visitado.
Analizamos 2.

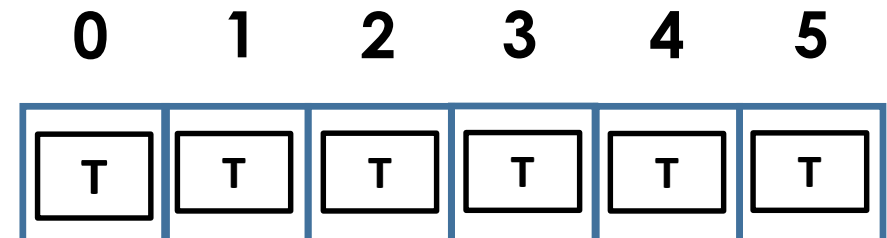
Lo marcamos como visitado

0	1	2	3	4	5
T	T	T	T	T	T

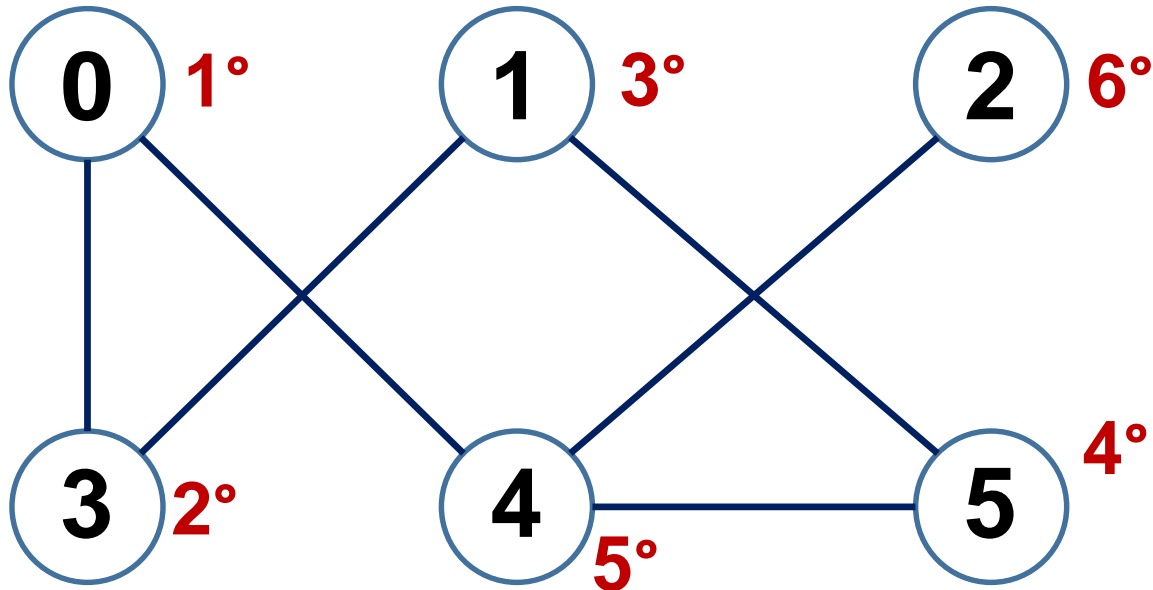
BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

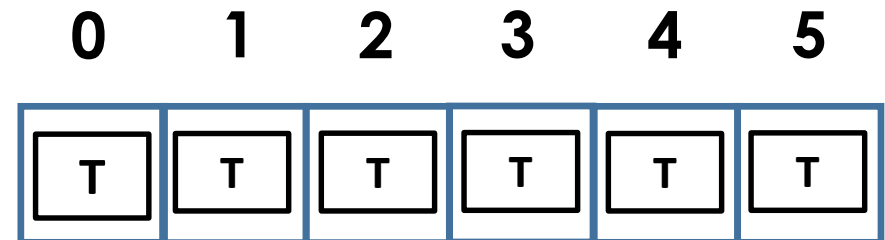


Adyacentes de 2: 4
4 ya visitado
No se puede continuar
recorrido en profundidad

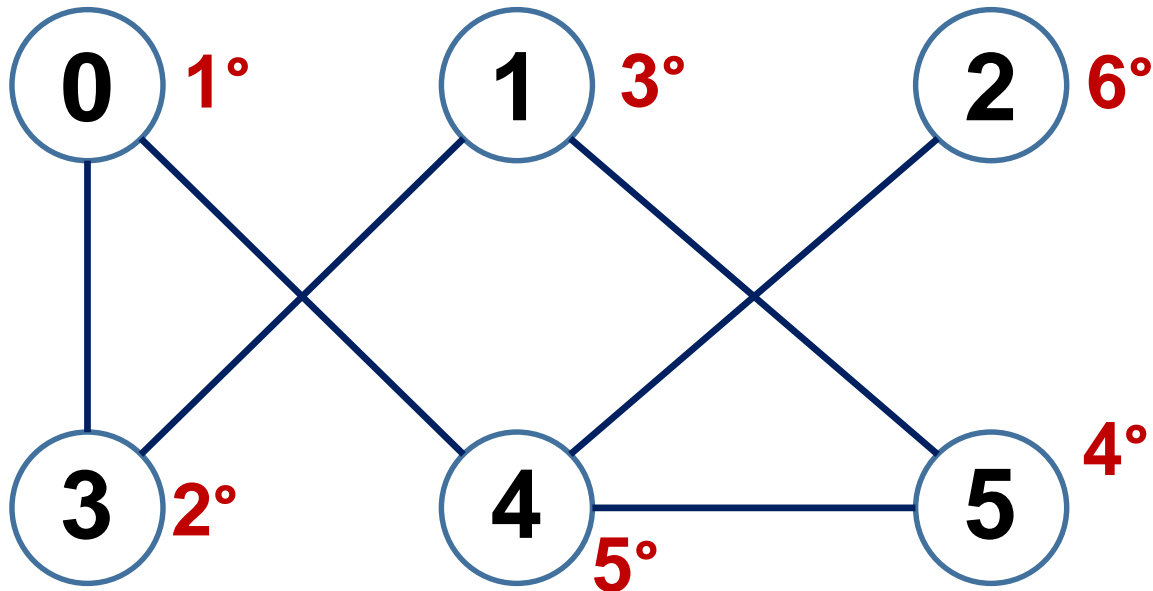
PILA (STACK) →

2
4
5
1
3
0

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

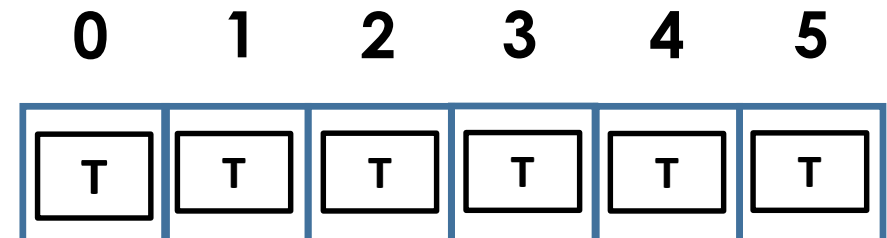


Entonces se “regresa”
Es decir sacamos a 2
de la pila.

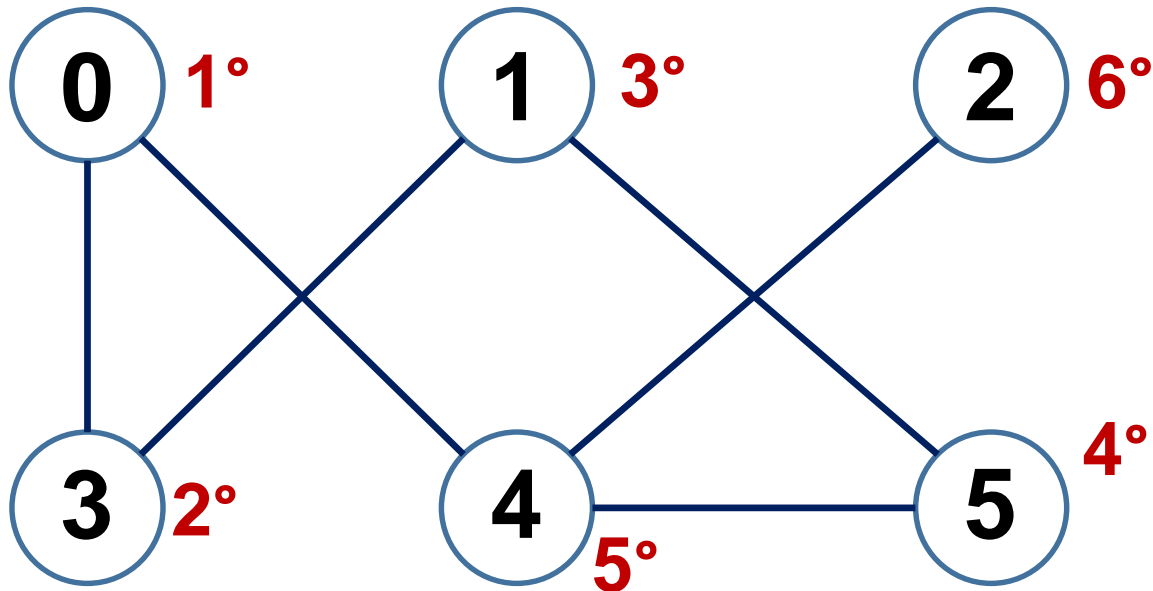
PILA
(STACK) →

2
4
5
1
3
0

Y también nos apoyamos en un arreglo
para “marcar” con un booleano (T o F)
cuando hemos visitado un nodo.
Cada posición representará un nodo:



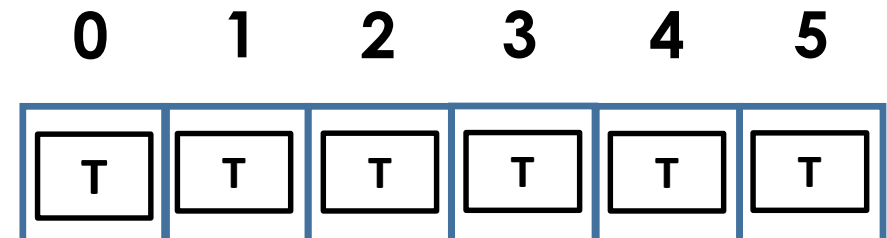
BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



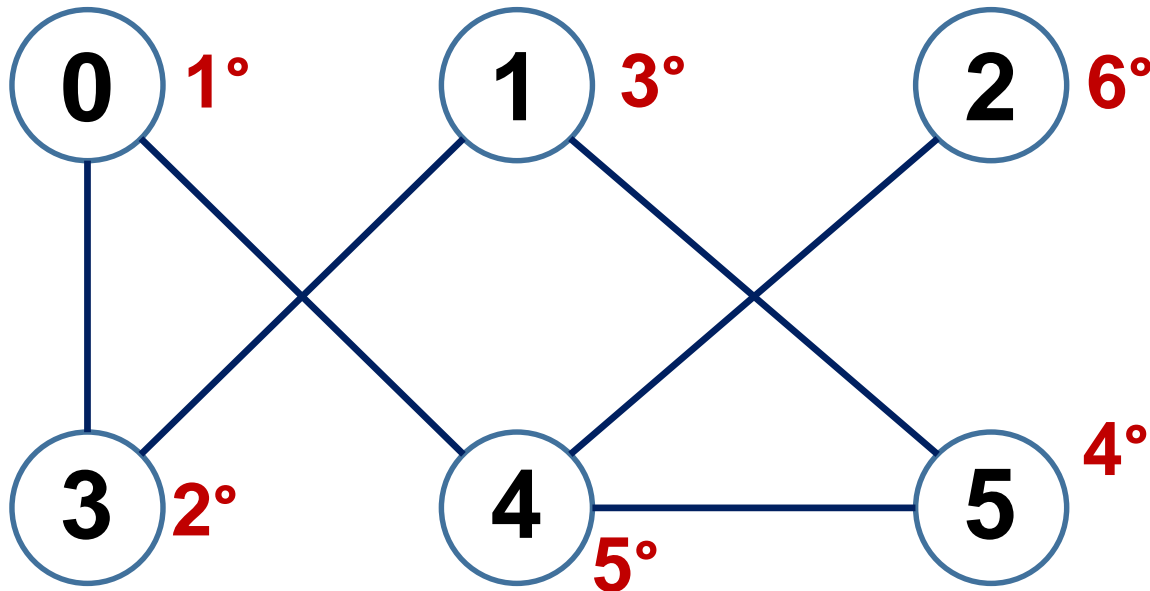
PILA
(STACK) →

4
5
1
3
0

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

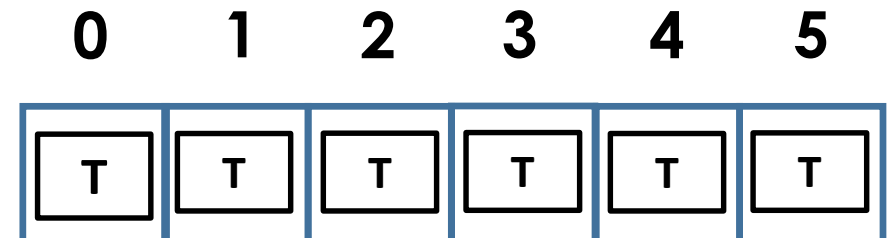


Adyacentes de 4: 0 , 2 , 5
0 ya visitado.
2 ya visitado.
5 ya visitado.
NO se puede seguir.
Se saca a 4 de la pila.

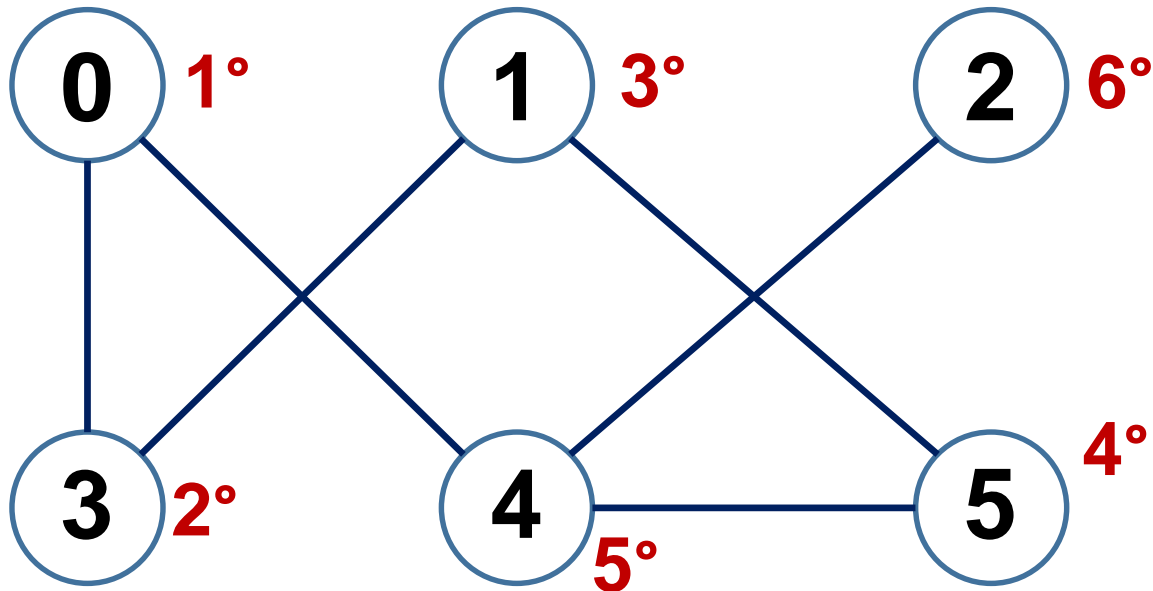
PILA (STACK) →

4
5
1
3
0

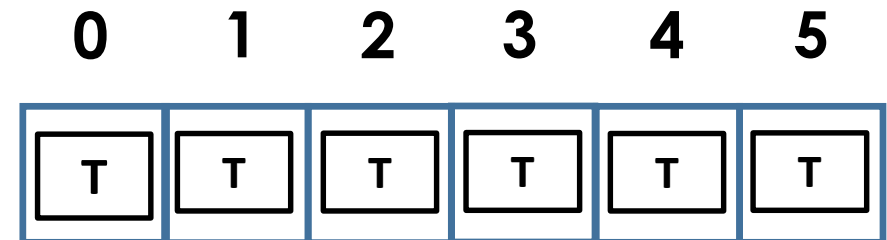
Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



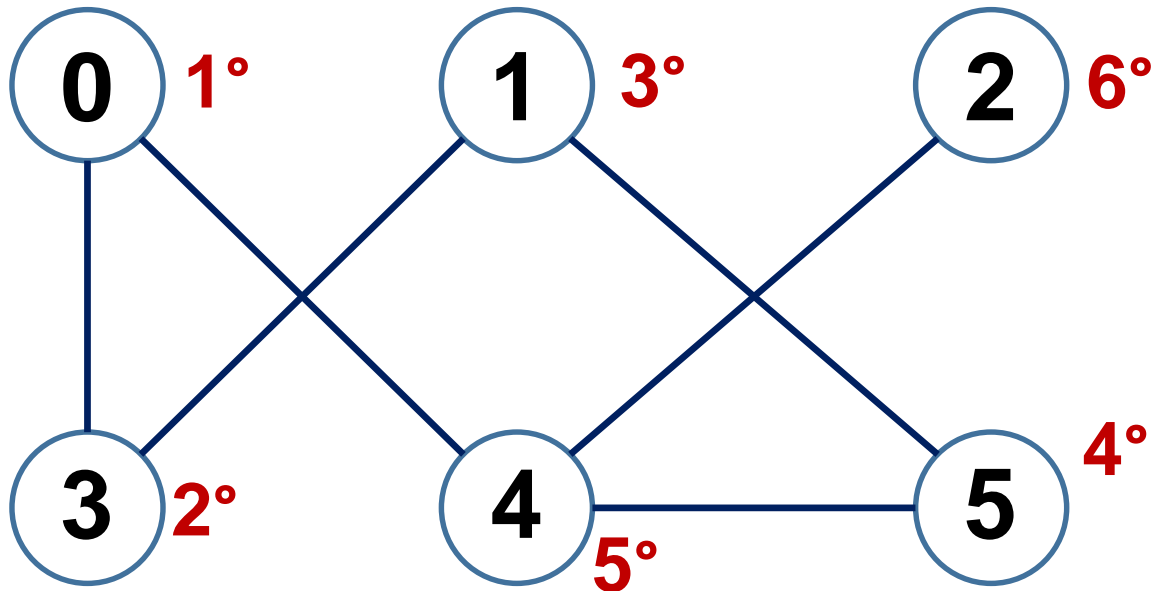
BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



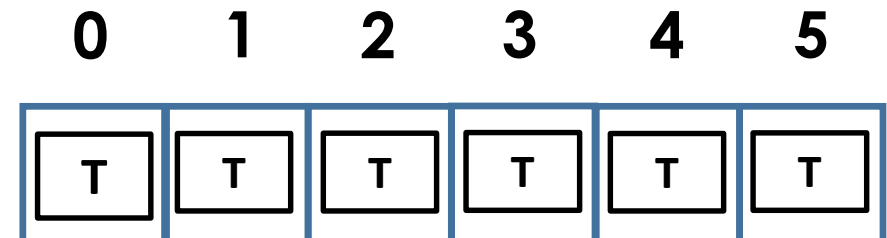
Adyacentes de 5: 1, 4
1 ya visitado.
4 ya visitado.

NO se puede seguir.
Se saca a 5 de la pila.

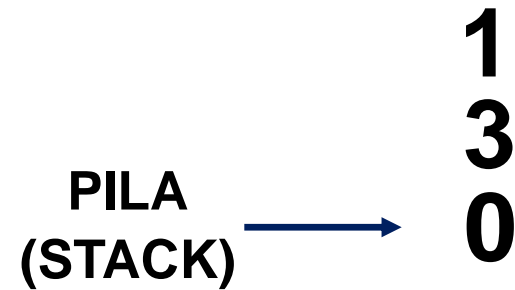
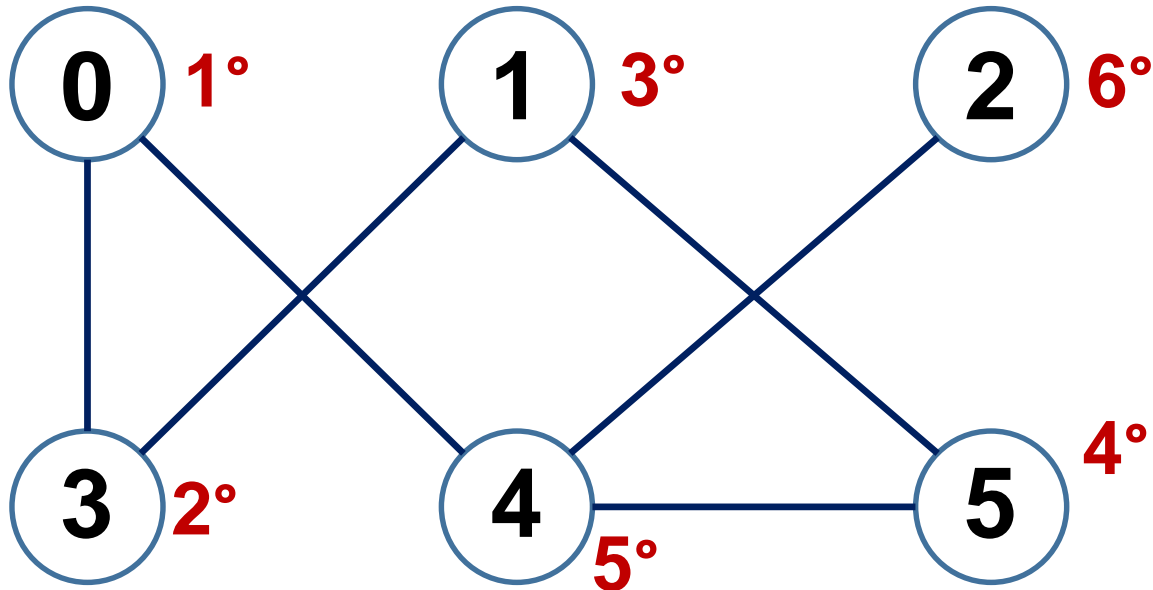
PILA (STACK) →

5
1
3
0

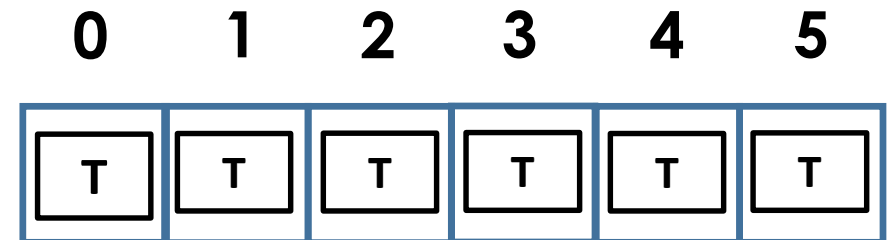
Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



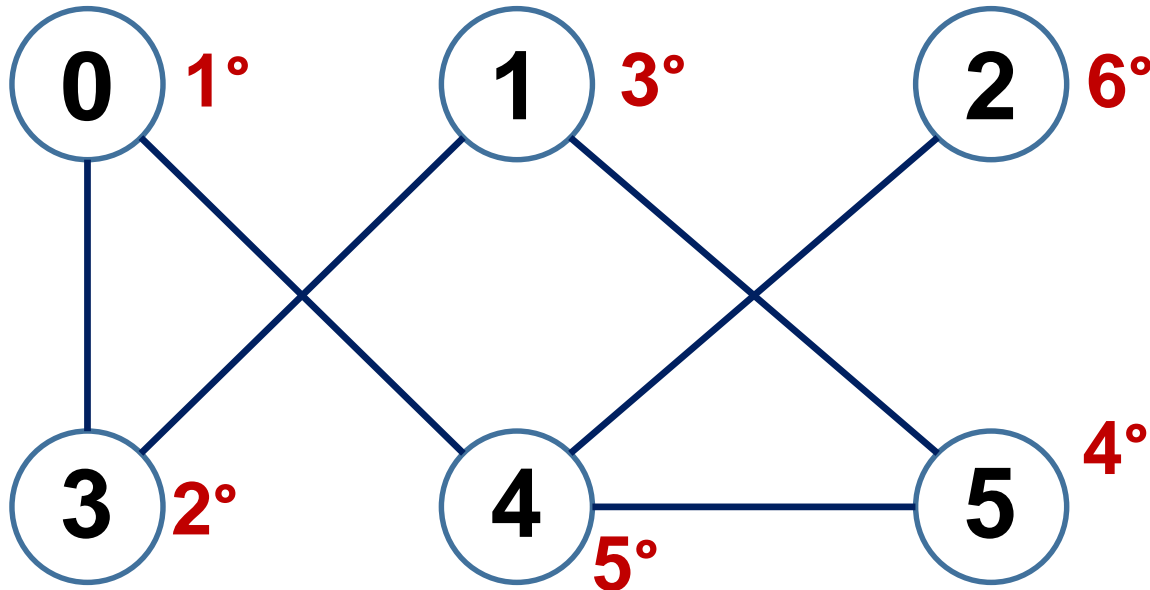
BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



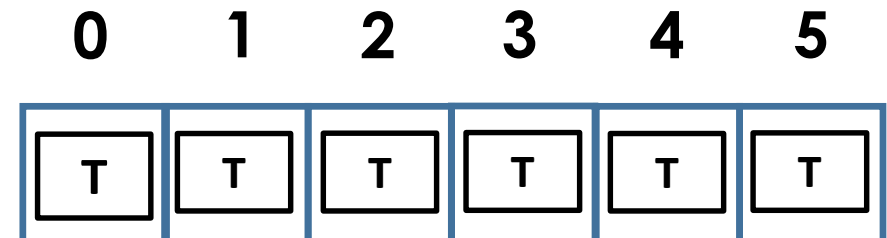
Adyacentes de 1: 3, 5
3 ya visitado.
5 ya visitado.

NO se puede seguir.
Se saca a 1 de la pila.

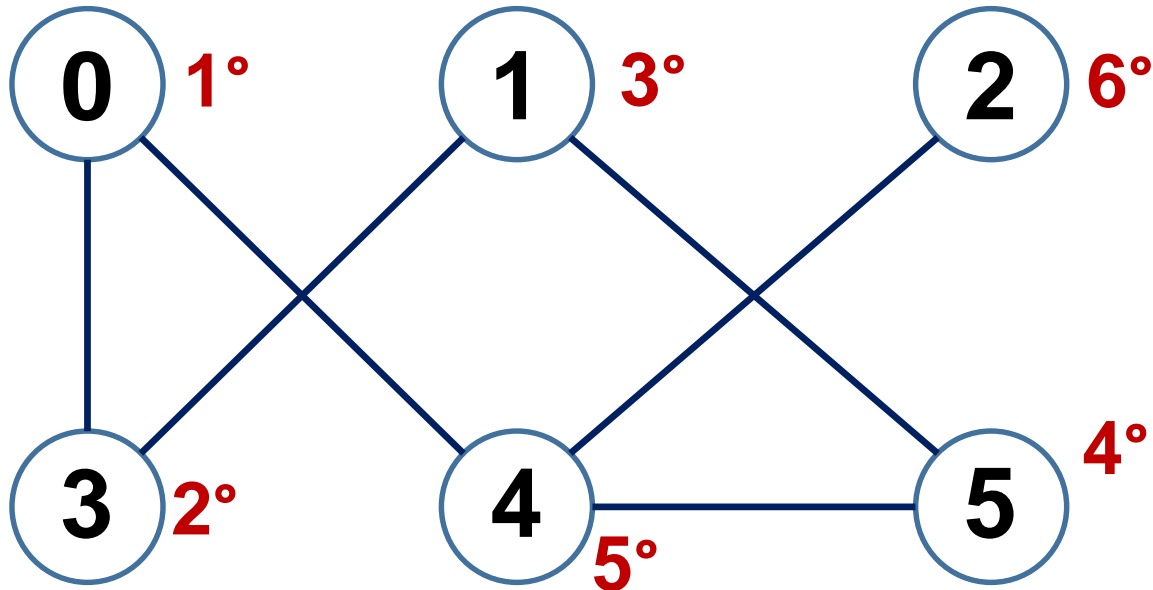
PILA
(STACK)

1
3
0

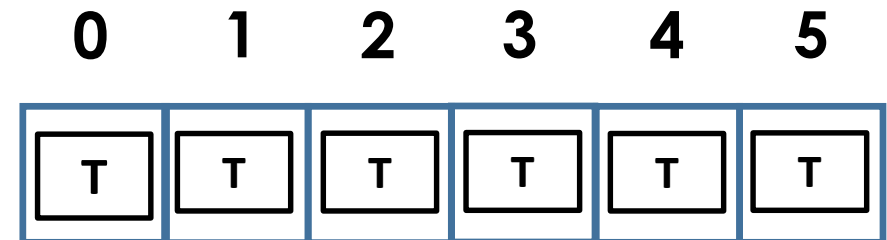
Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



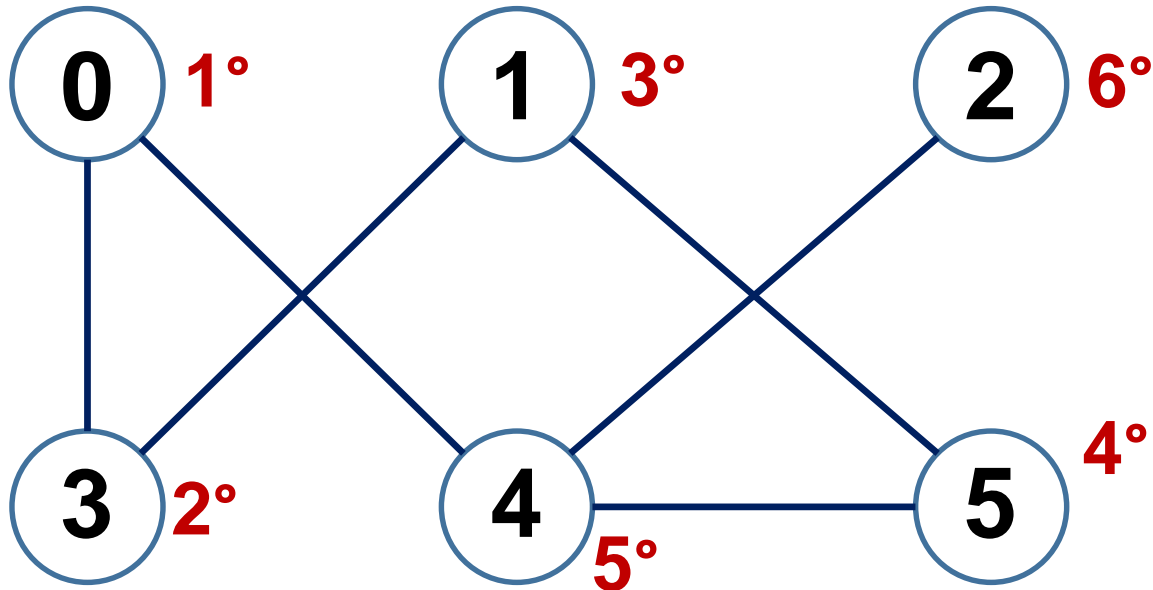
BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

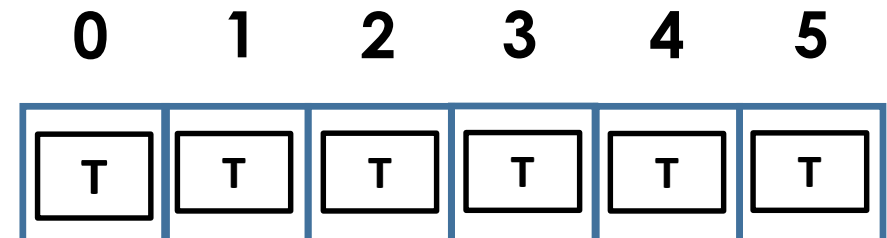


Adyacentes de 3: 0 , 1
0 ya visitado.
1 ya visitado.

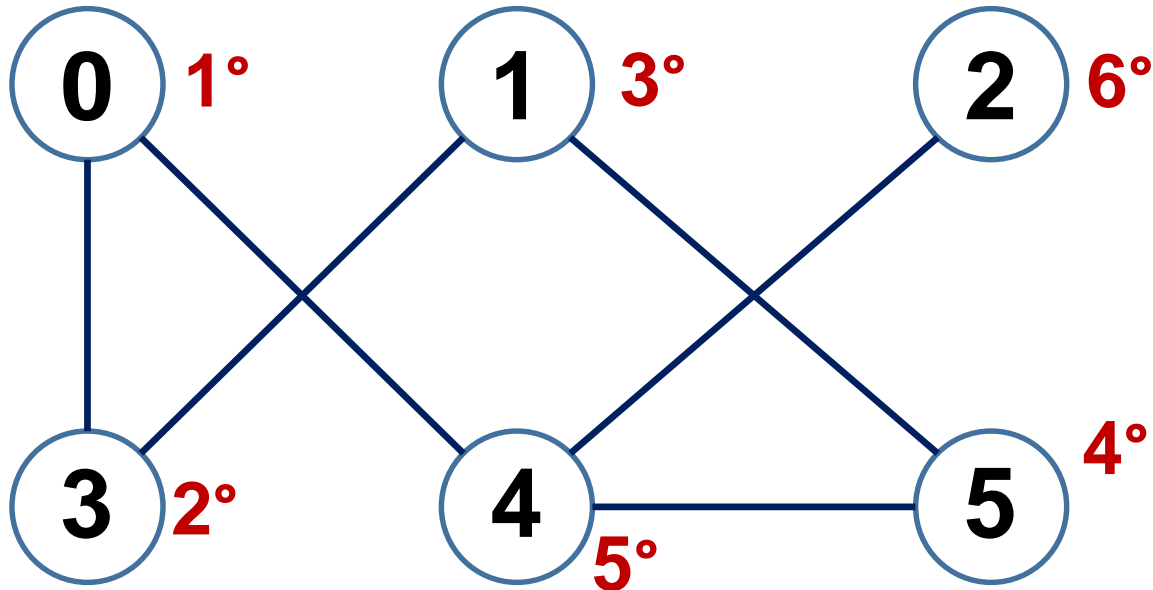
NO se puede seguir.
Se saca a 3 de la pila.

PILA (STACK) → 3
0

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

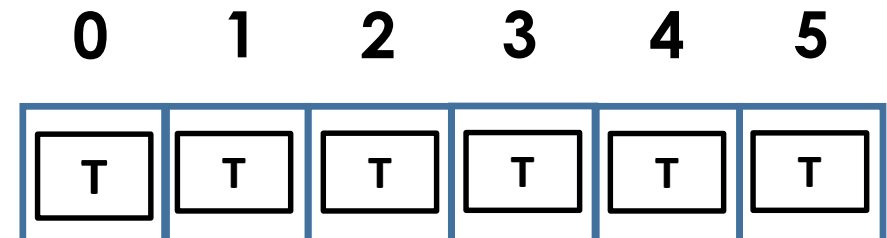


BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

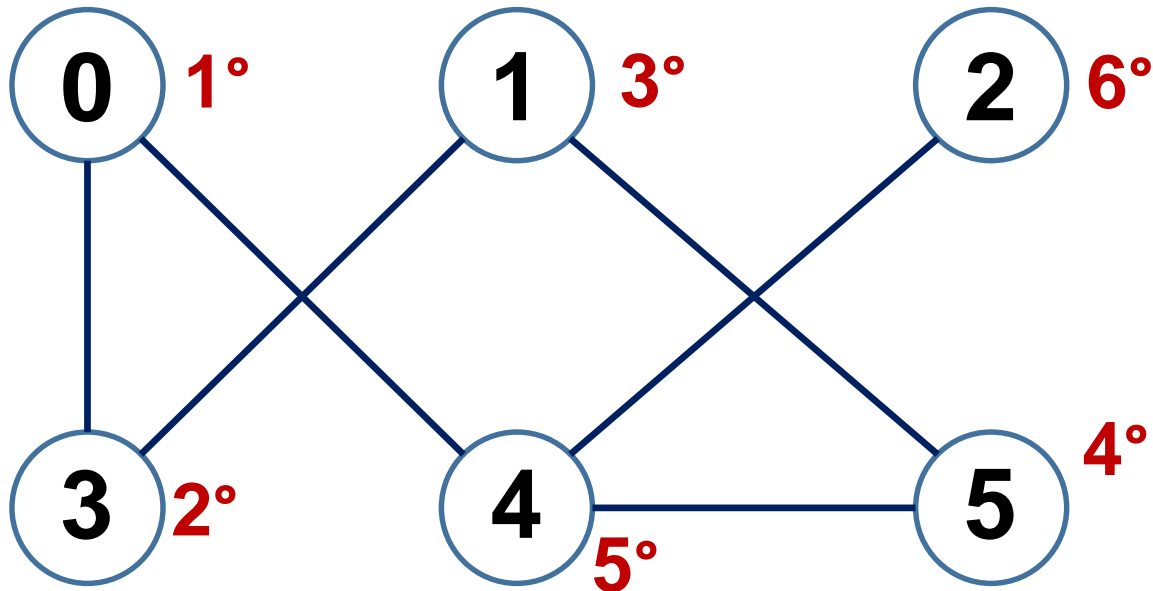


PILA (STACK) → 0

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

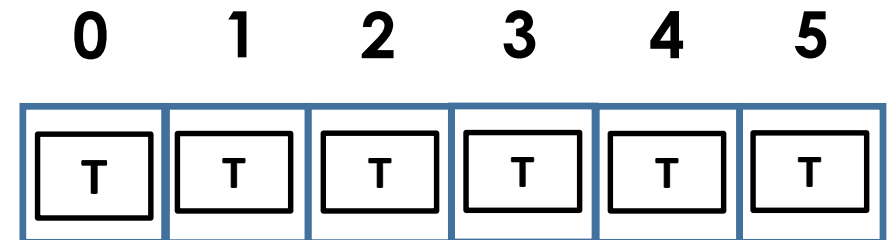


PILA (STACK) → 0

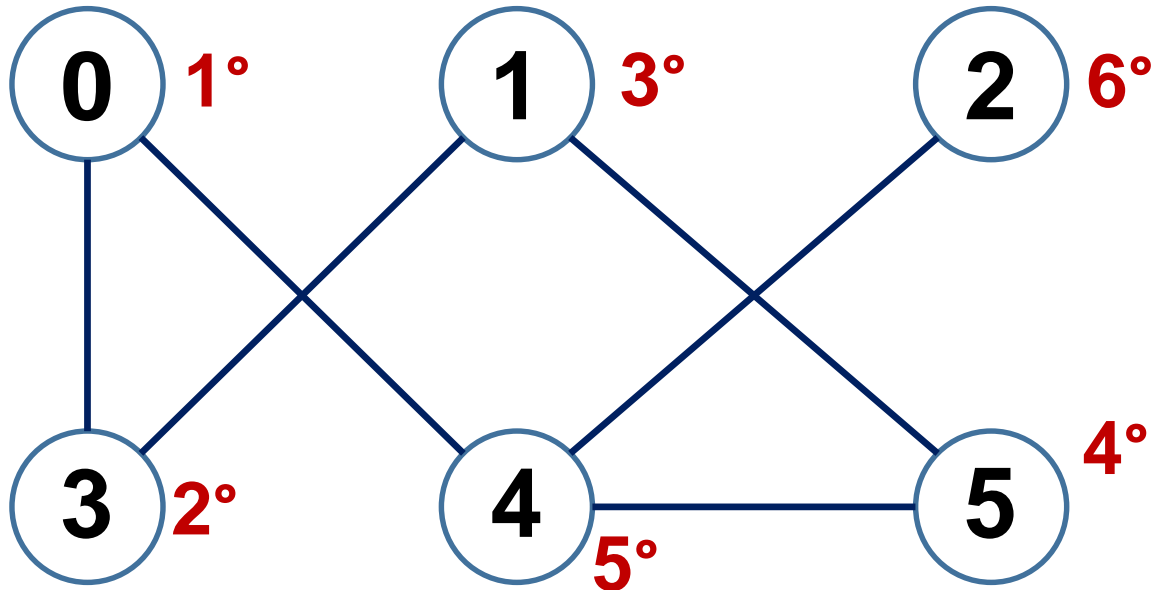
Adyacentes de 0: 3, 4
3 ya visitado.
4 ya visitado.

NO se puede seguir.
Se saca a 0 de la pila.

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:

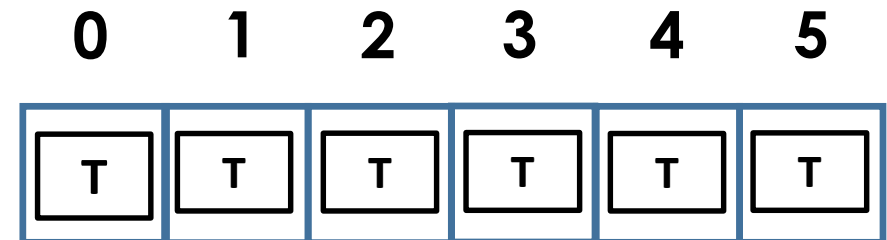


BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)

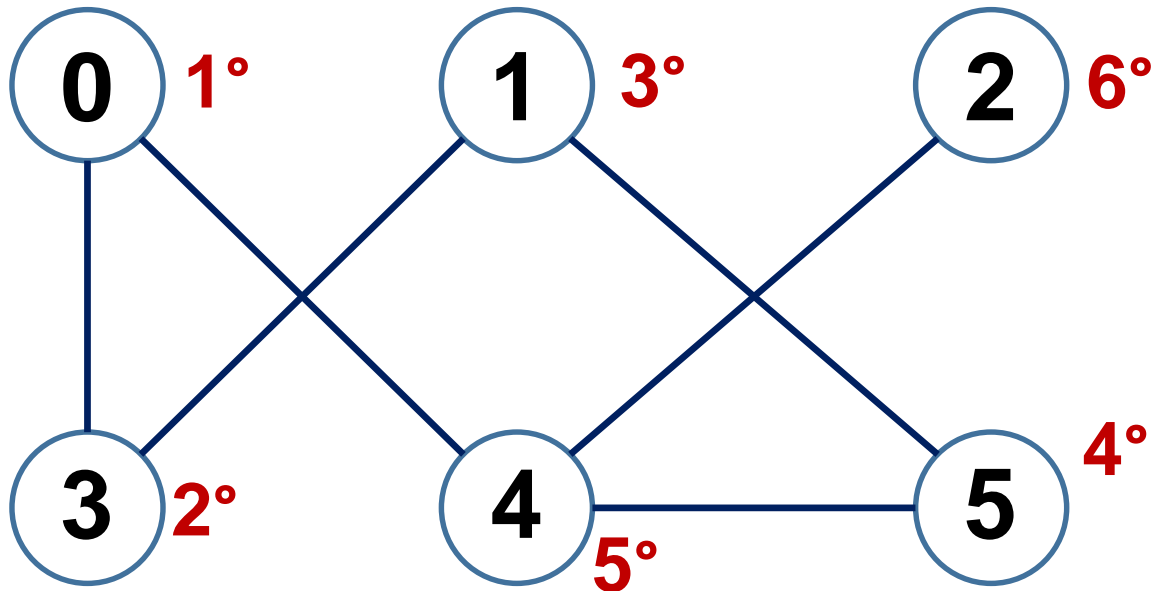


PILA
(STACK) →

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



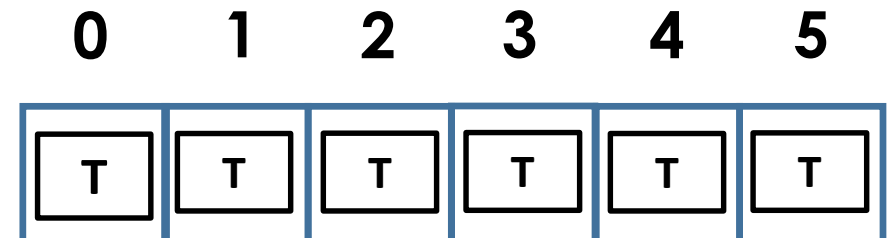
BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



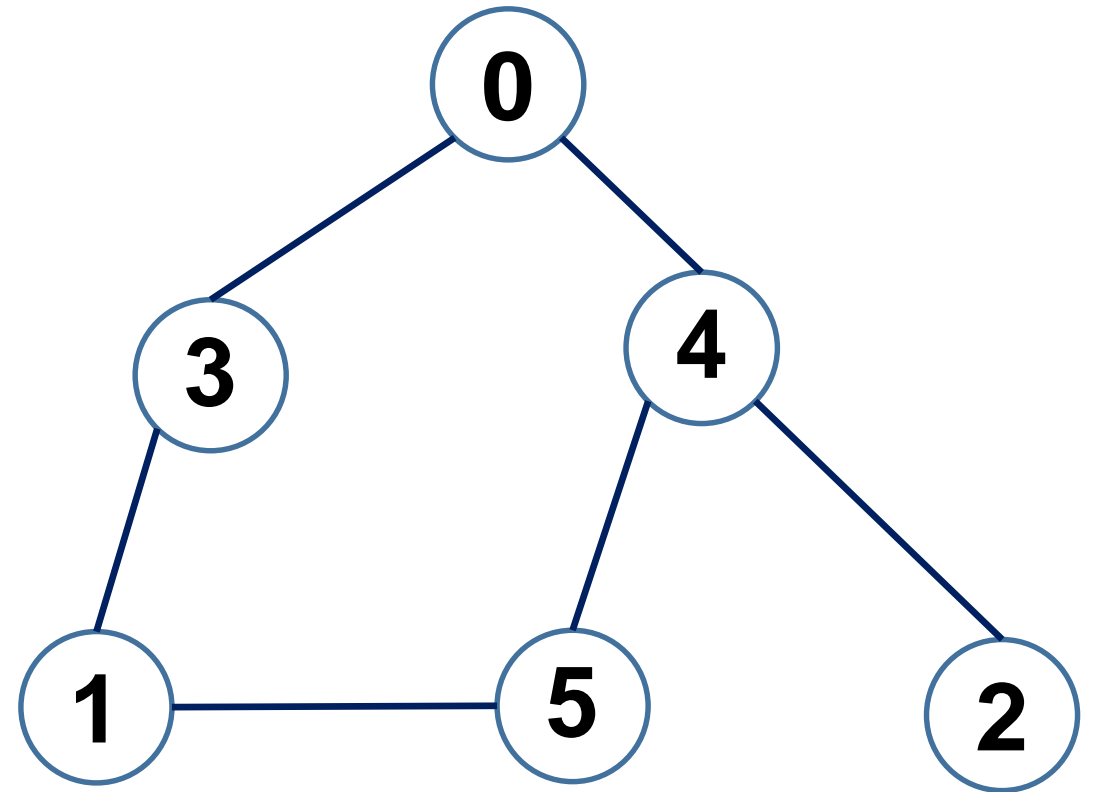
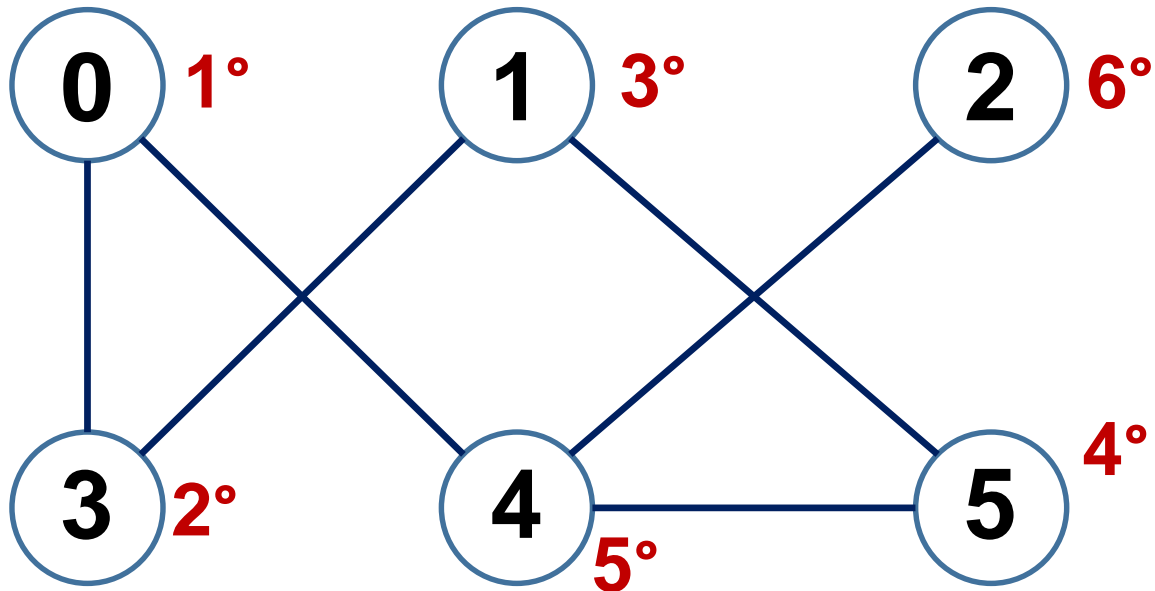
PILA (STACK) →

PILA VACÍA
Termina la ejecución

Y también nos apoyamos en un arreglo para “marcar” con un booleano (T o F) cuando hemos visitado un nodo. Cada posición representará un nodo:



BÚSQUEDA EN PROFUNDIDAD DFS (Depth First Search)



RECORRIDO DEL GRAFO EN PROFUNDIDAD DFS

0 - 3 - 1 - 5 - 4 - 2

Sobre su aplicación

Si luego de ejecutar 1 vez el BFS o el DFS,
alguno de los nodos no fue visitado,
el grafo es NO CONEXO.

Si se desea saber cuántas componentes conexas tiene un grafo,
basta con contar la cantidad de veces
que se llamó al algoritmo de recorrido
hasta que todos los nodos fueran visitados

----- FIN DEL DOCUMENTO