

Ejercicio Supernode (2017 1c 1p e1)

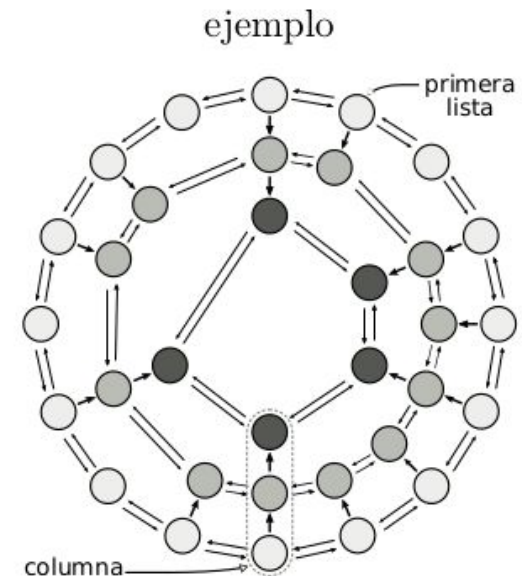
Ej. 1. (35 puntos)

Sea la siguiente estructura de listas doblemente enlazadas encadenadas entre si.

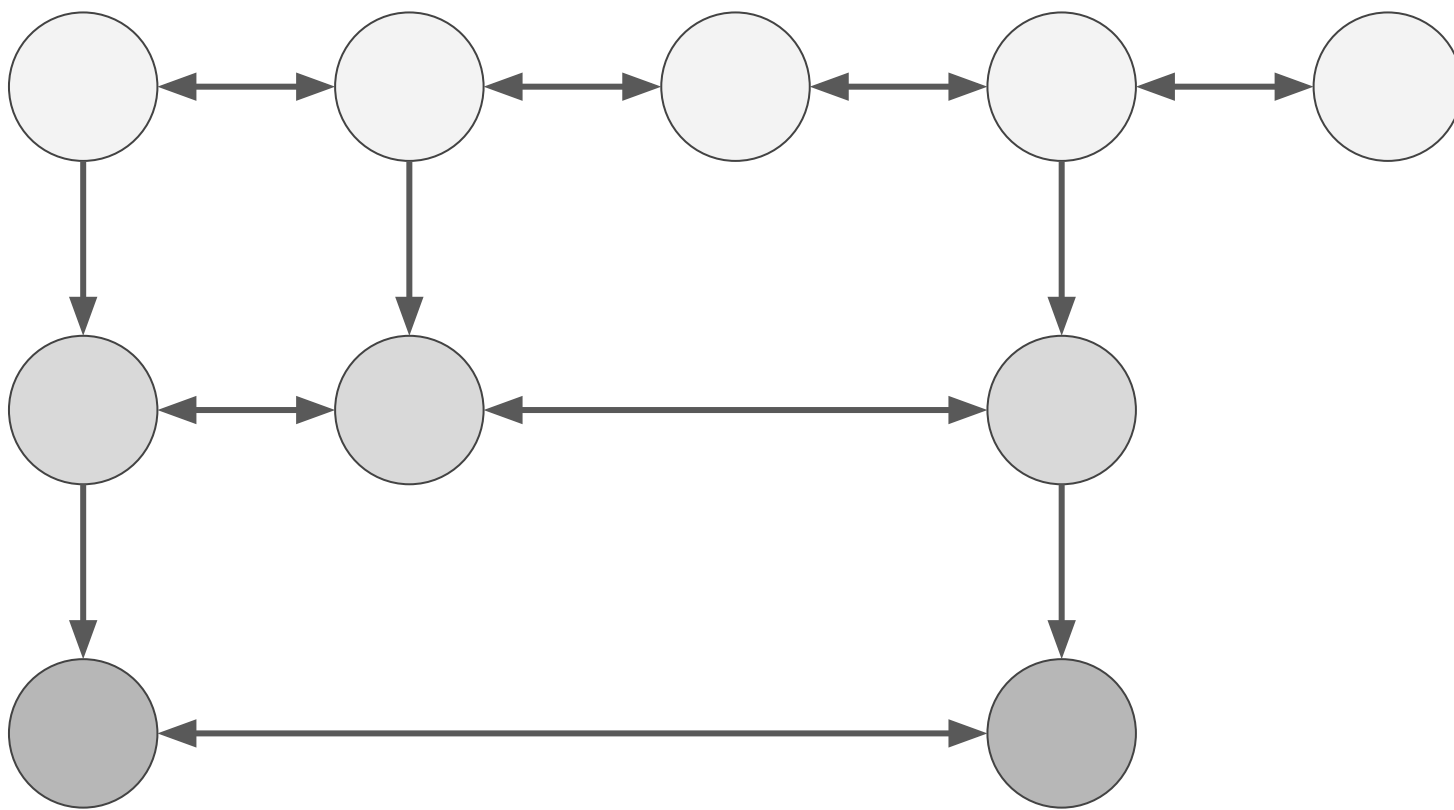
```
struct supernode {  
    supernode* abajo,  
    supernode* derecha,  
    supernode* izquierda,  
    int dato };
```

- todos los nodos pertenecen a una lista doblemente enlazada
- todos los nodos son referenciados desde algun otro nodo en otra lista (excepto en la primera)
- todas las listas respetan el orden de los nodos que las apuntan

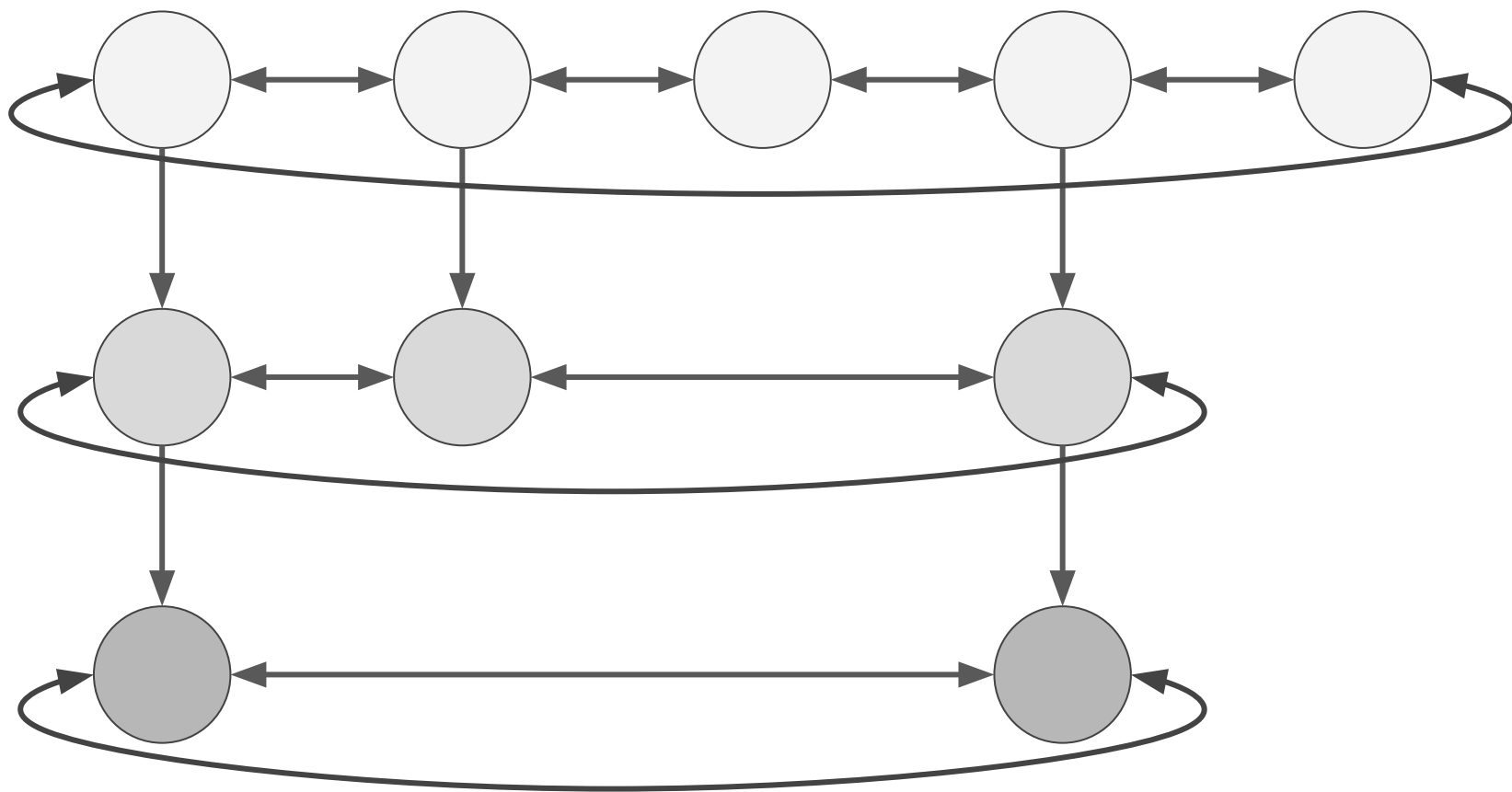
Implementar en ASM las siguientes funciones.



Supernode



Supernode

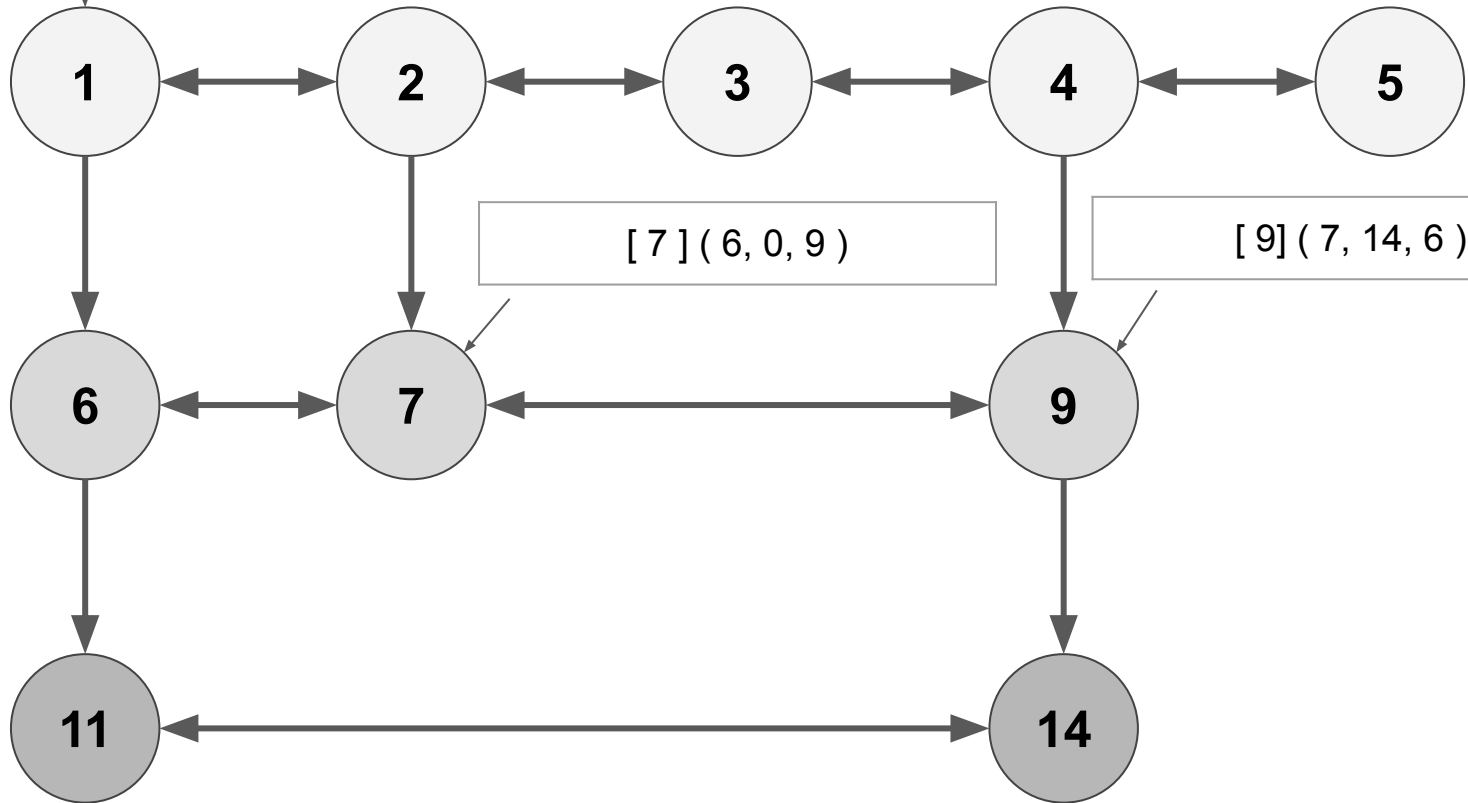


Representación

[ID] (IZQ, ABA, DER)

```
struct supernode {  
    supernode* abajo,  
    supernode* derecha,  
    supernode* izquierda,  
    int dato };
```

[1] (5, 6, 2)



[7] (6, 0, 9)

[9] (7, 14, 6)

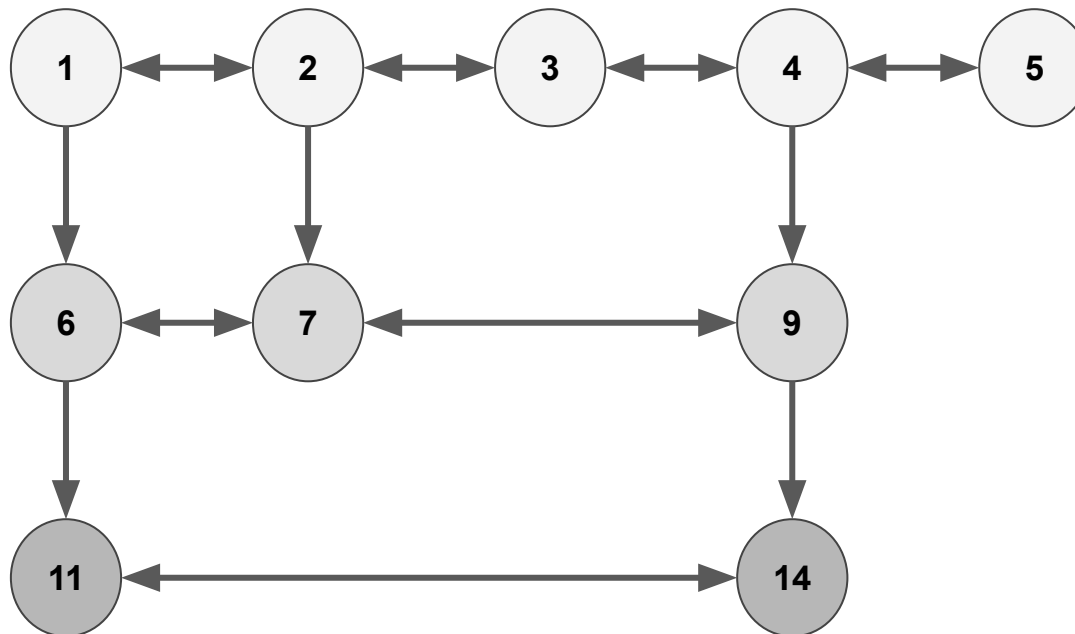
Representación

Setup

[01] (05, 06, 02) [02] (01, 07, 03) [03] (02, 00, 04) [04] (03, 09, 05) [05] (04, 00, 01)

[06] (09, 11, 07) [07] (06, 00, 09) [09] (07, 14, 06)

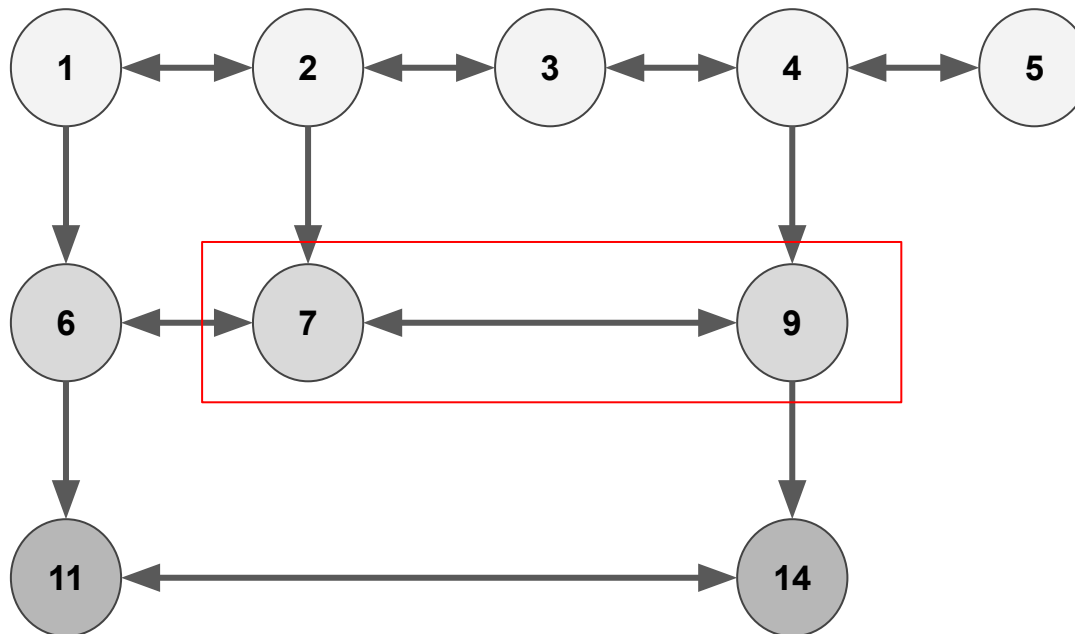
[11] (14, 00, 14) [14] (11, 00, 11)



Representación

Setup

[01] (05, 06, 02) [02] (01, 07, 03) [03] (02, 00, 04) [04] (03, 09, 05) [05] (04, 00, 01)
[06] (09, 11, 07) [07] (06, 00, 09) [09] (07, 14, 06)
[11] (14, 00, 14) [14] (11, 00, 11)



Metodología

```
int main(void)
{
    ...
    borrar_columna(X);
    ...
}
```

main()



```
void borrar_columna(X)
{
    ...
    borrar_nodo(Y);
    ...
}
```

borrar_columna()



```
void borrar_nodo(Y)
{
    ...
}
```

borrar_nodo()



Metodología

```
int main(void)
{
    ...
    borrar_columna(X);
    ...
}
```

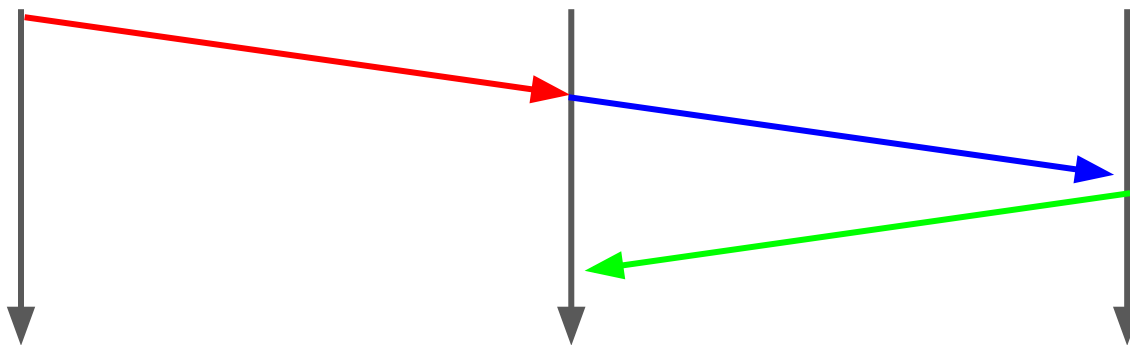
```
void borrar_columna(X)
{
    ...
    borrar_nodo(Y);
    ...
}
```

```
void borrar_nodo(Y)
{
    ...
    return;
}
```

main()

borrar_columna()

borrar_nodo()

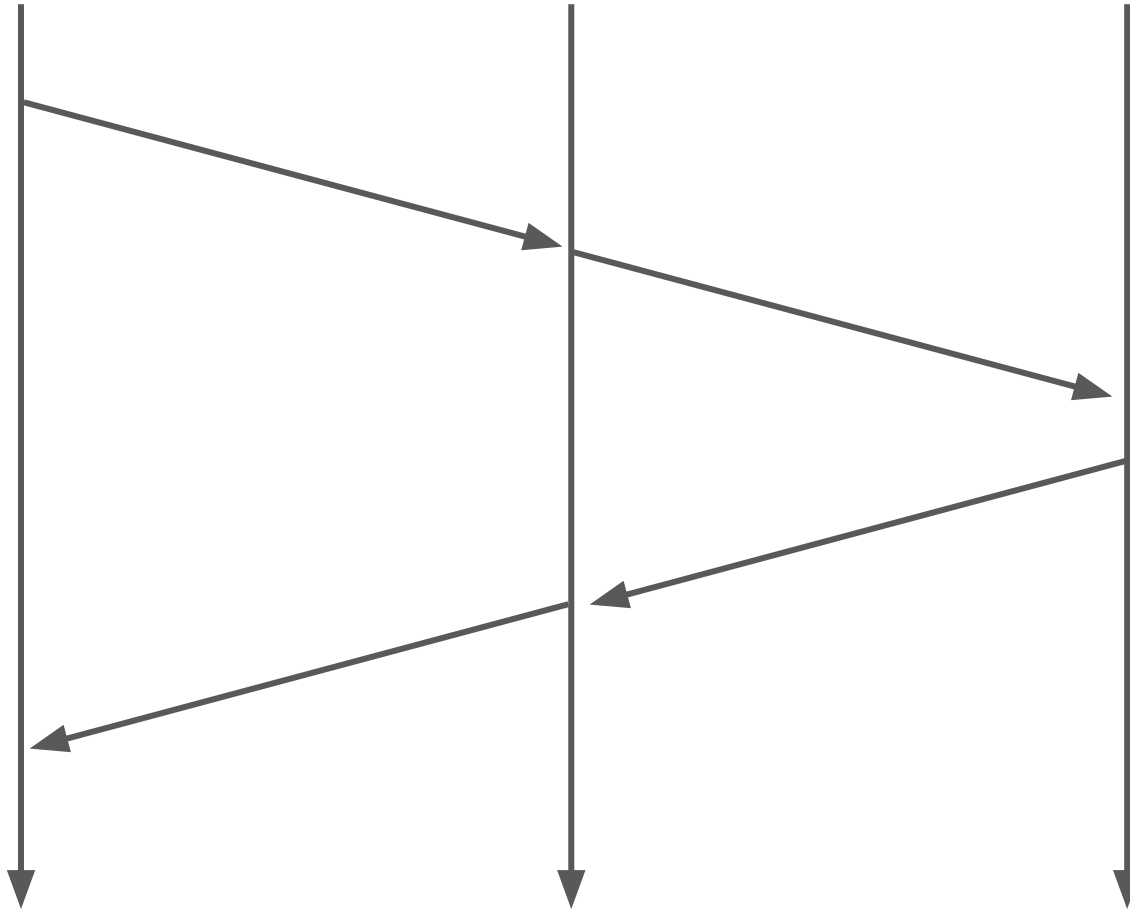


Metodología

main()

borrar_columna()

borrar_nodo()

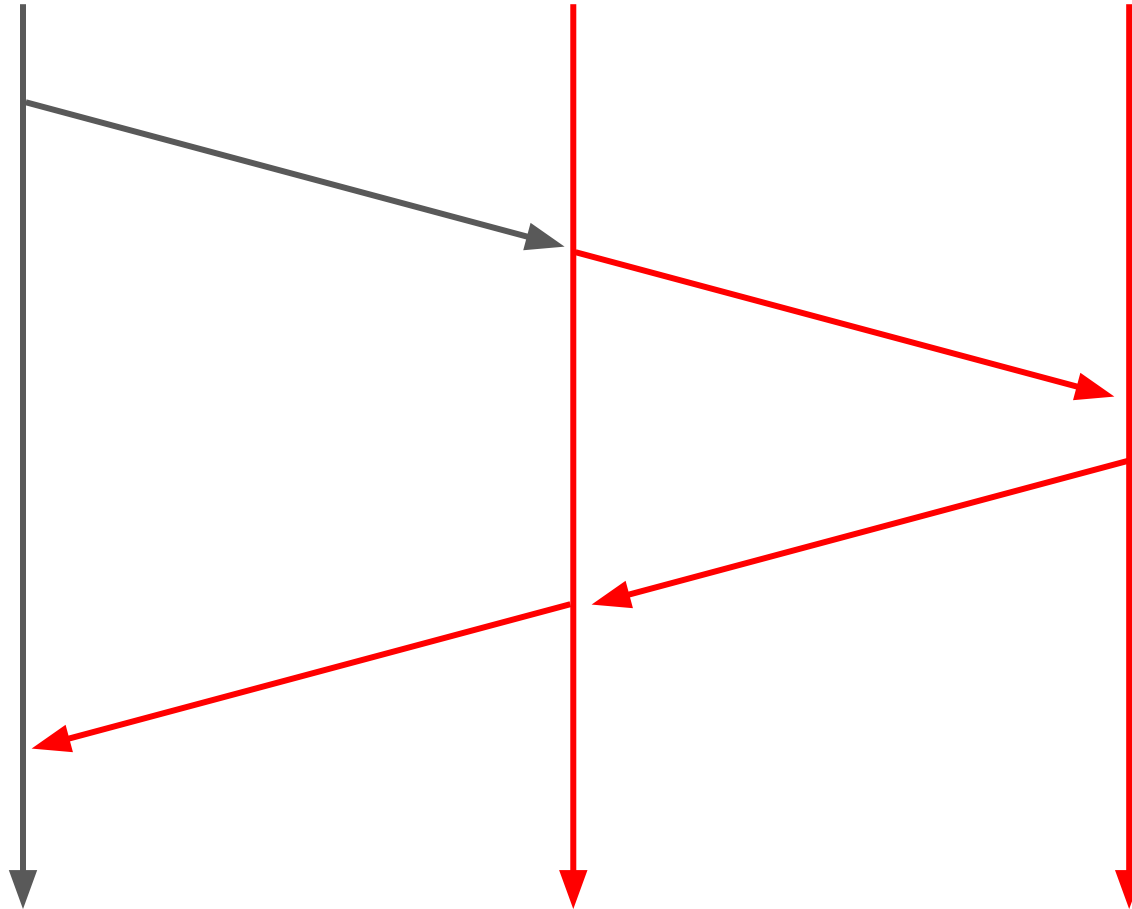


Metodología

main()

borrar_columna_C()

borrar_nodo_C()

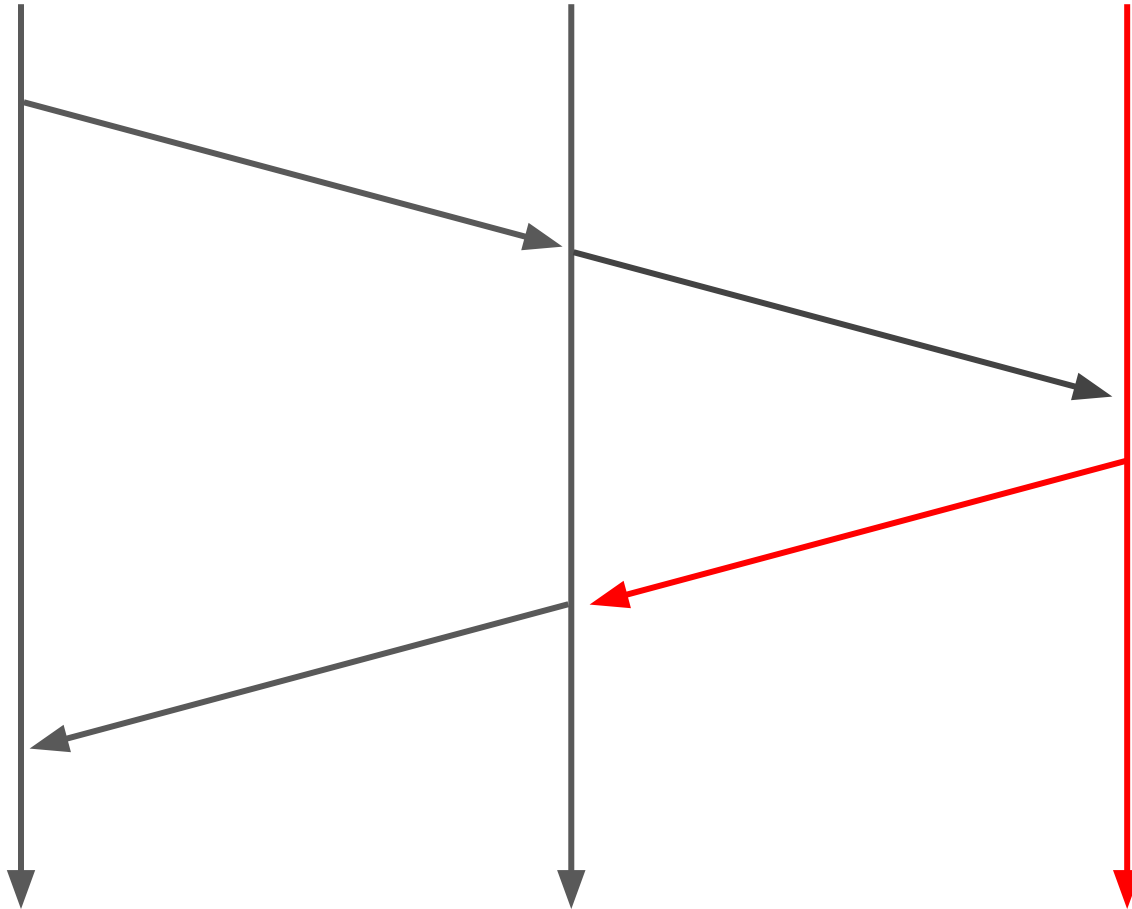


Metodología

main()

borrar_columna_C()

borrar_nodo_ASM()

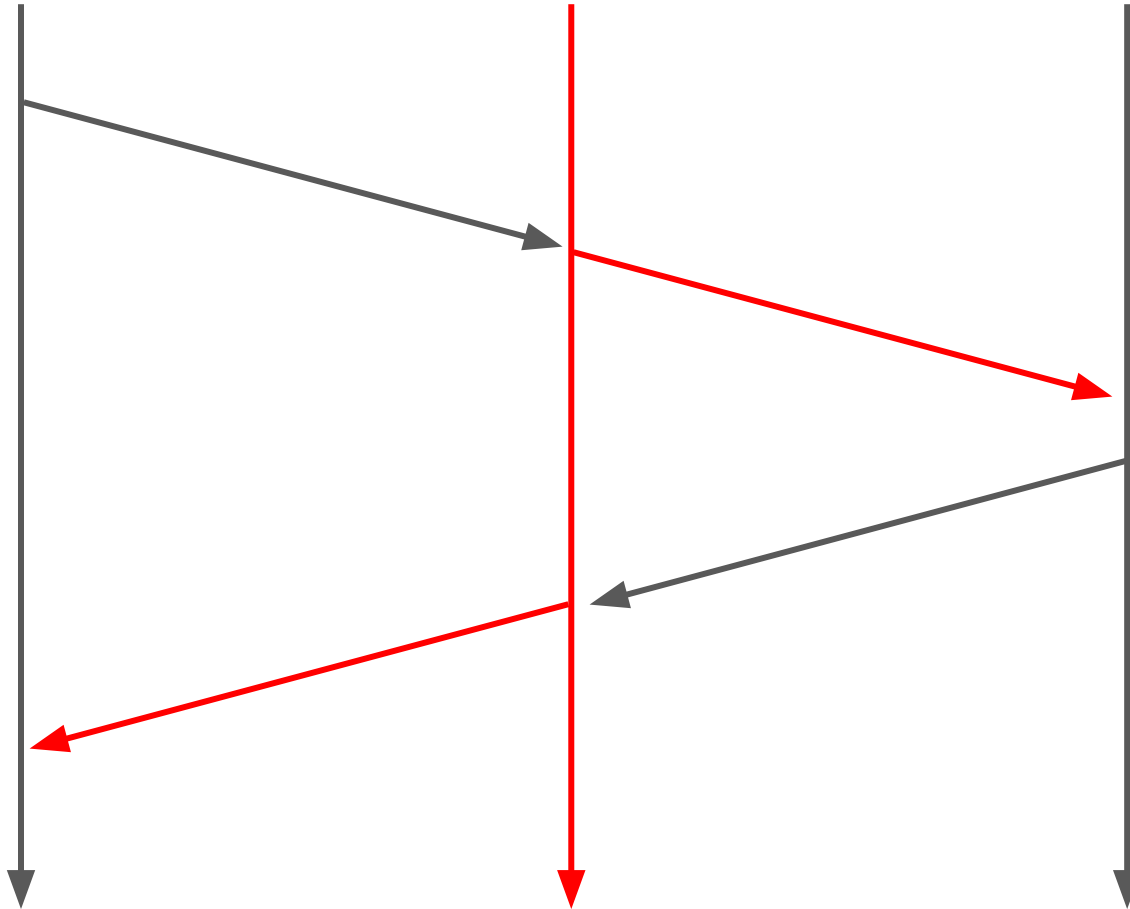


Metodología

main()

borrar_columna_ASM()

borrar_nodo_C()

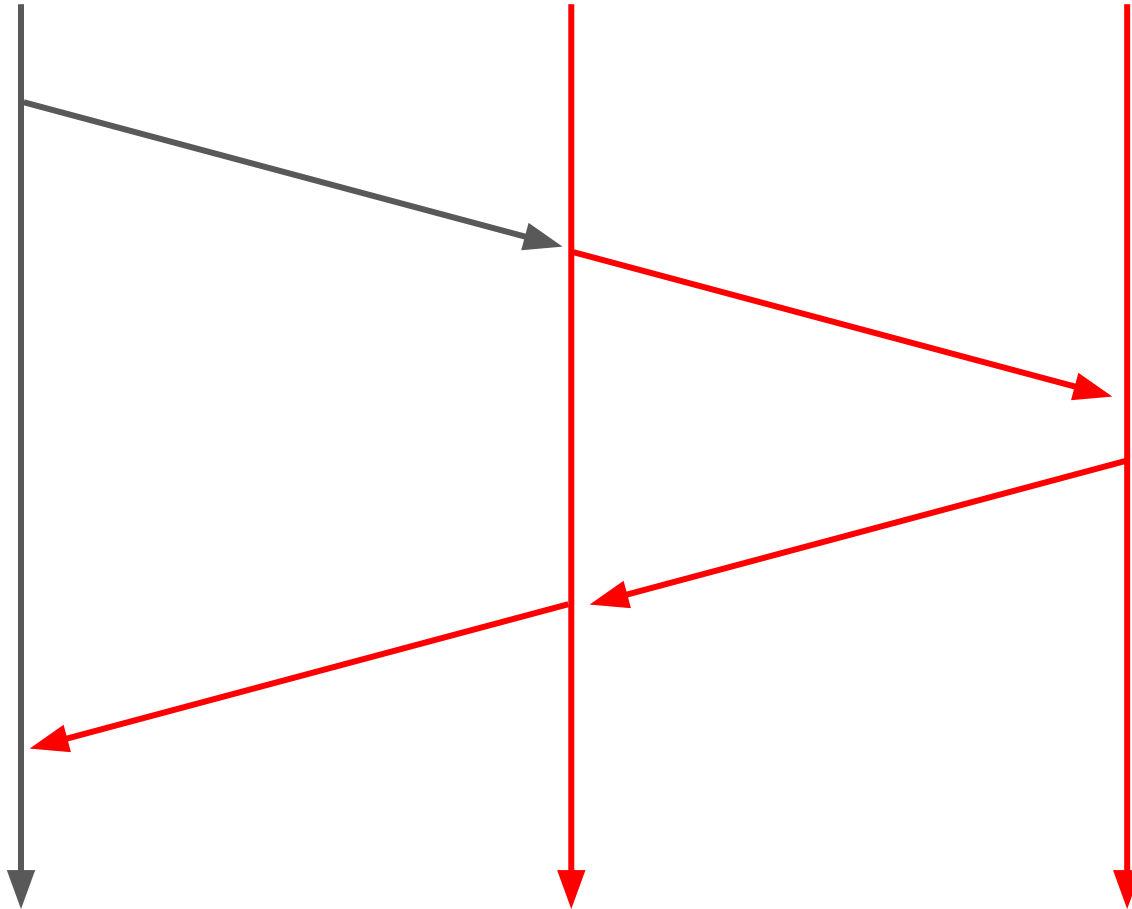


Metodología

main()

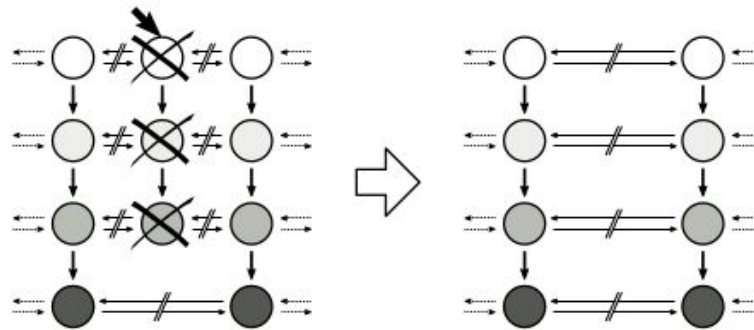
borrar_columna_ASM()

borrar_nodo_ASM()

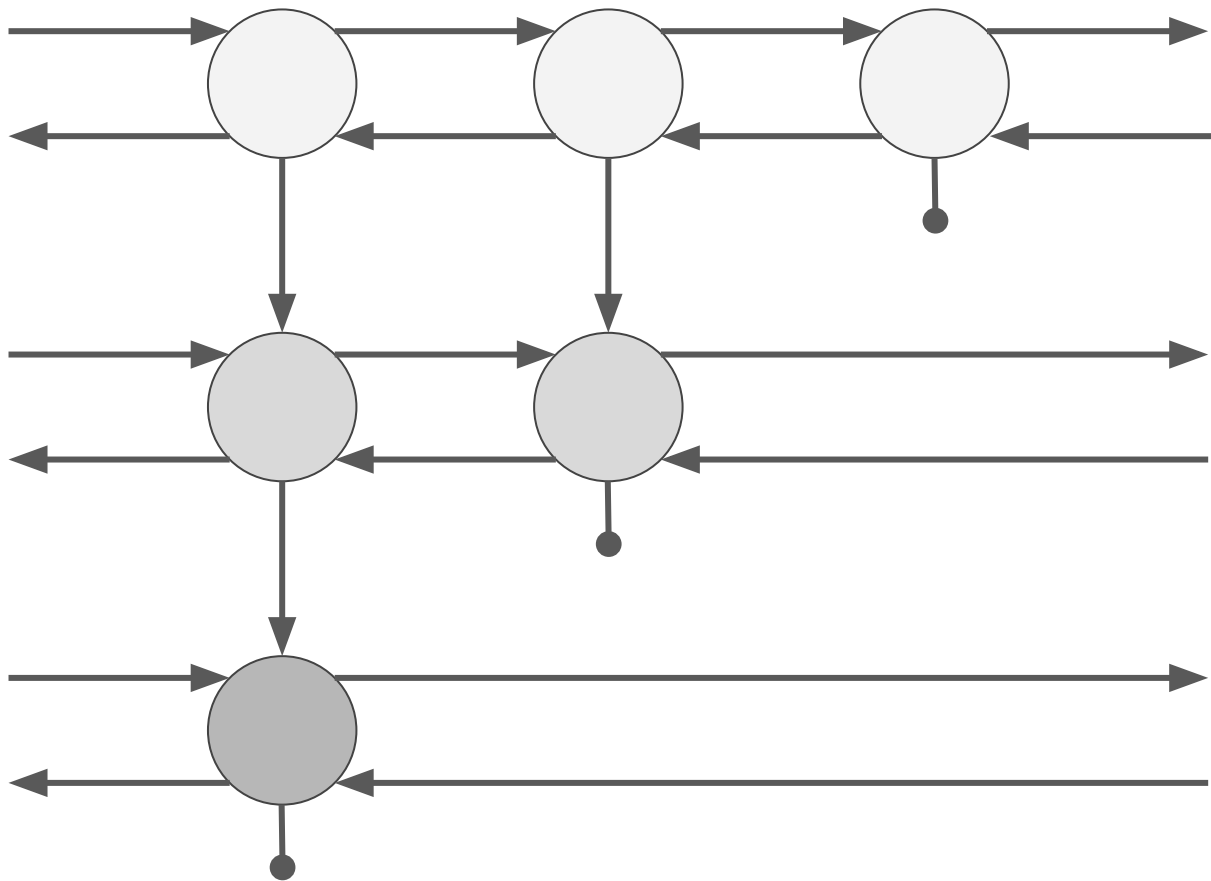


Borrar columna

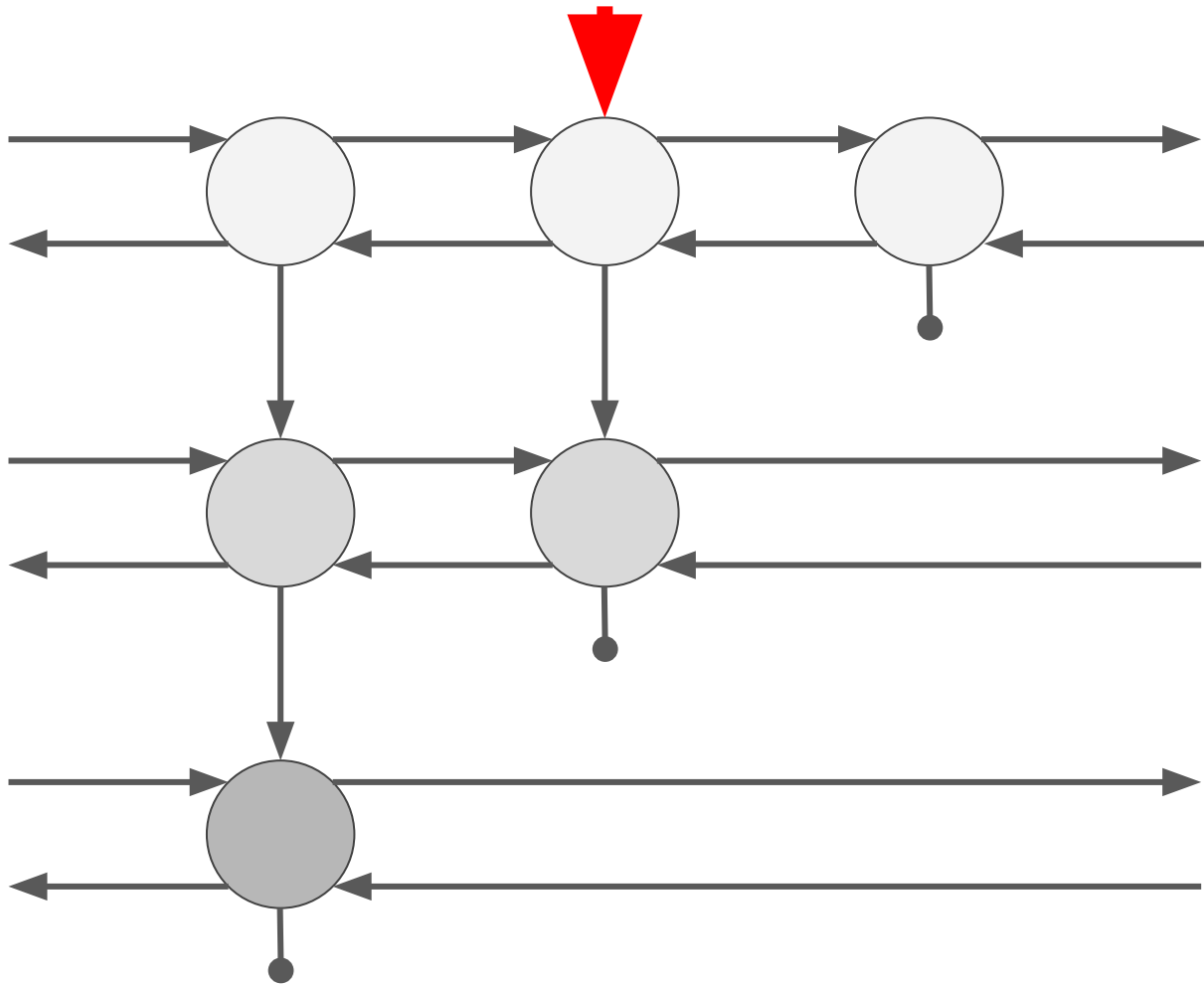
- a. `void borrar_columna(supernode** sn):` Dada un doble puntero a nodo dentro de la primera lista, borra una columna de nodos. Modifica el doble puntero dejando un nodo valido de la primer lista.



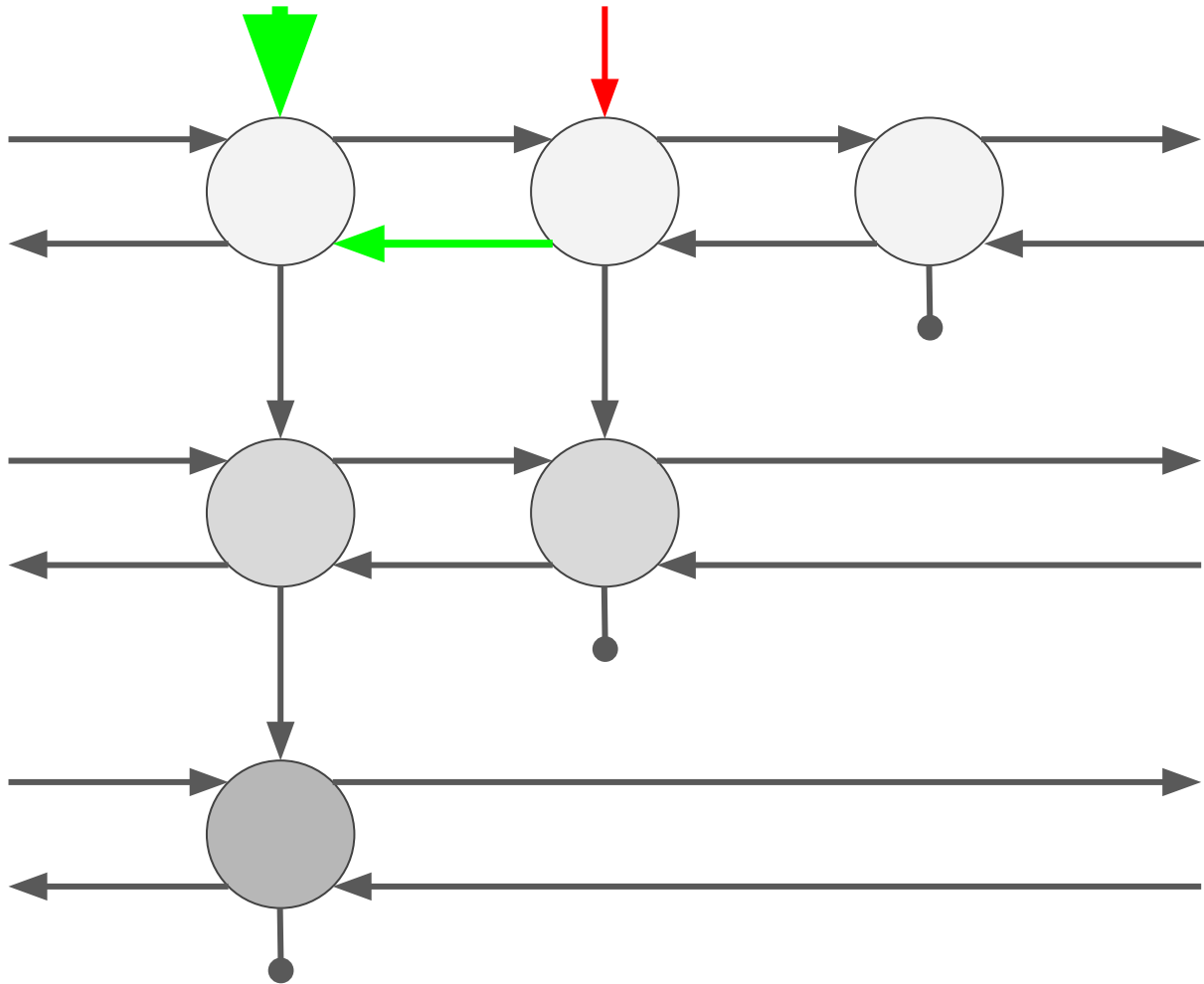
Borrar columna



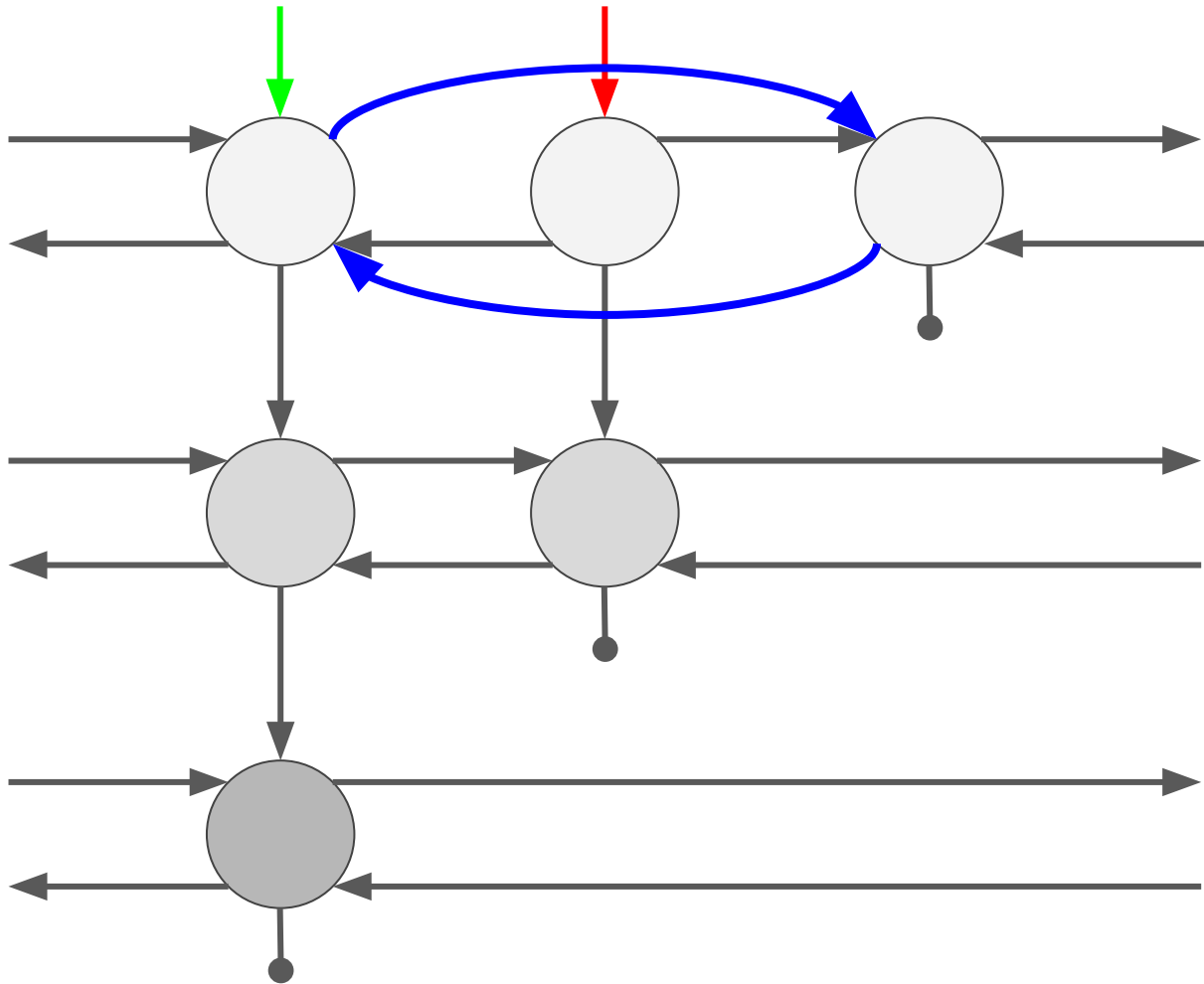
Borrar columna



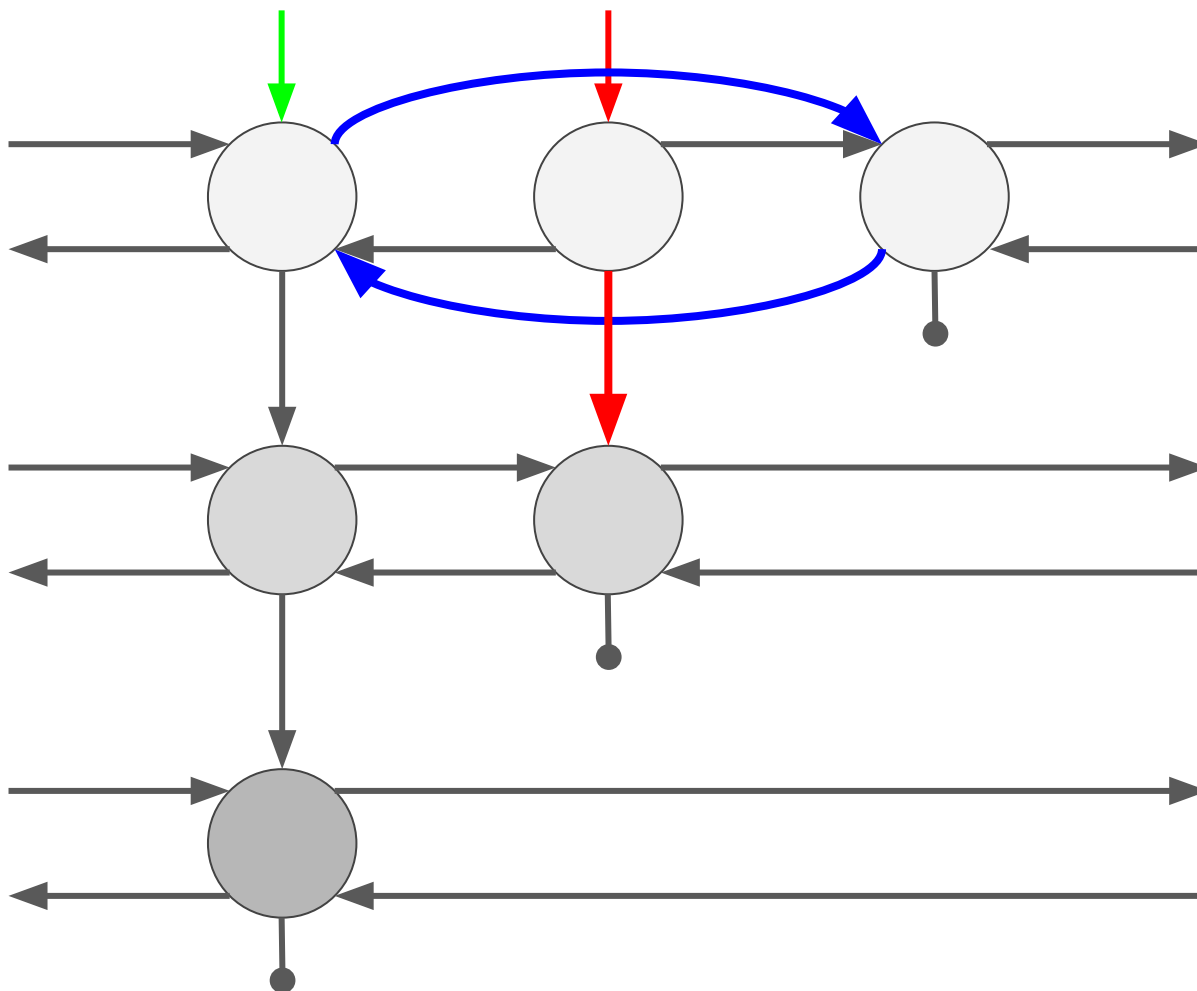
Borrar columna



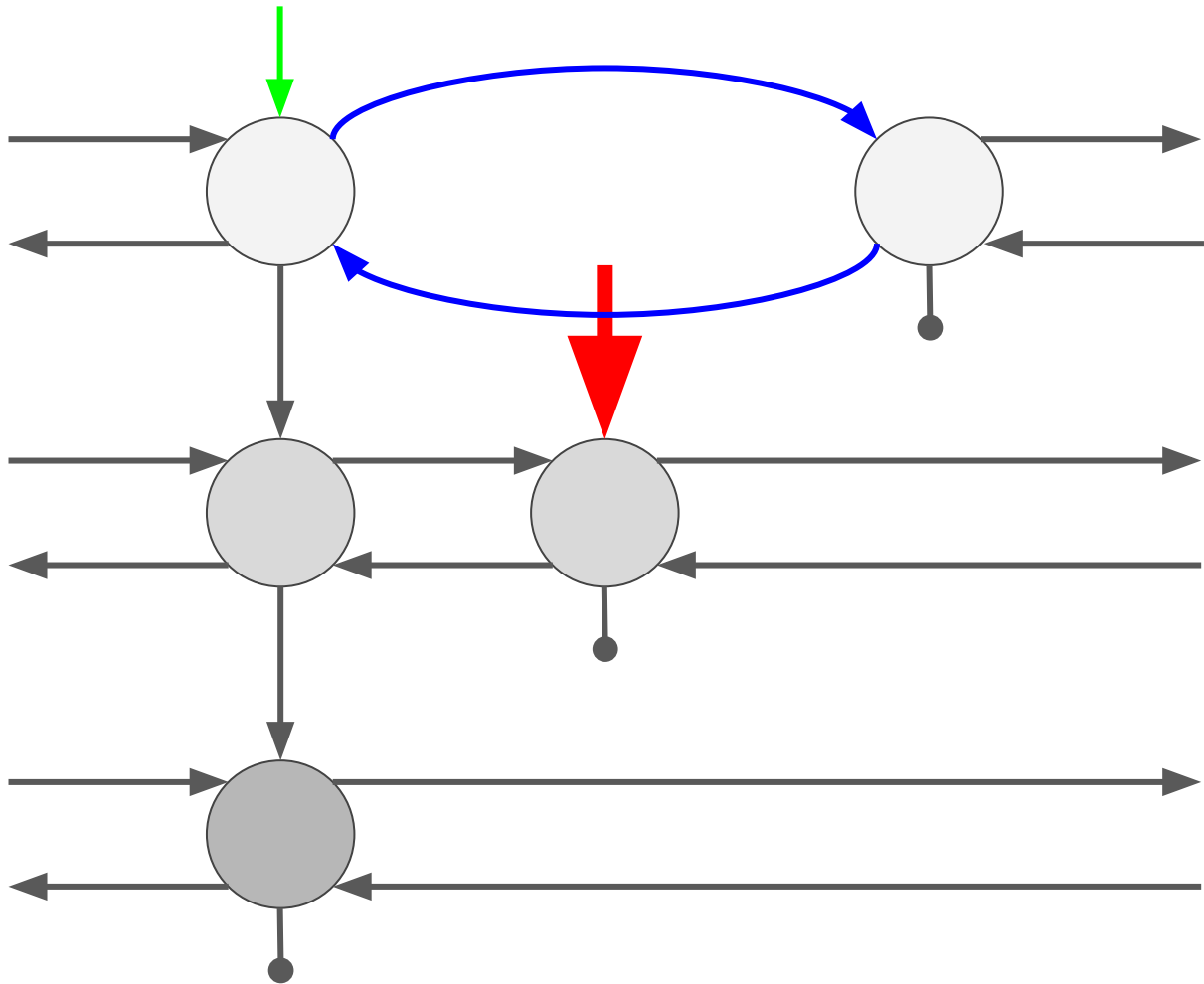
Borrar columna



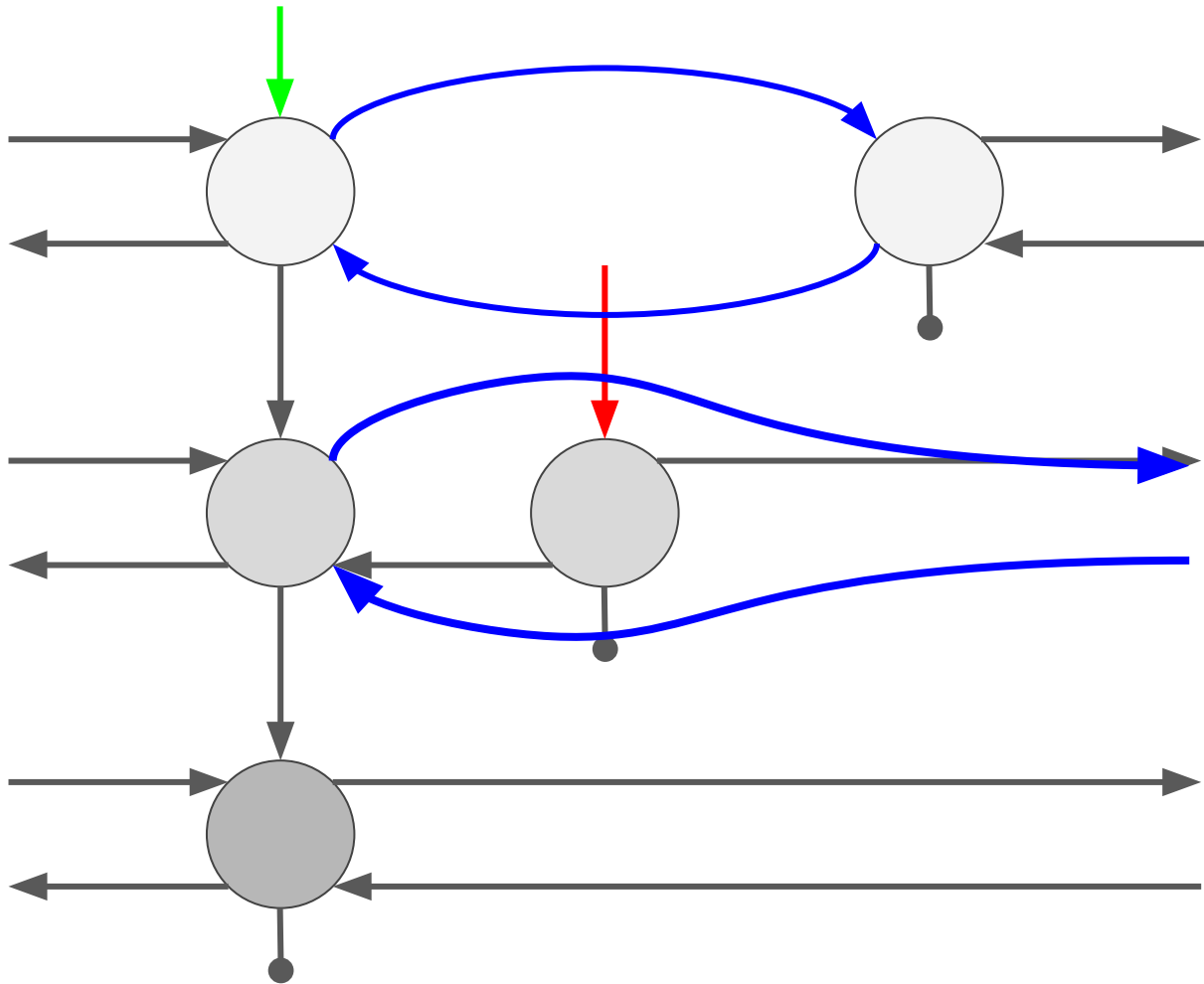
Borrar columna



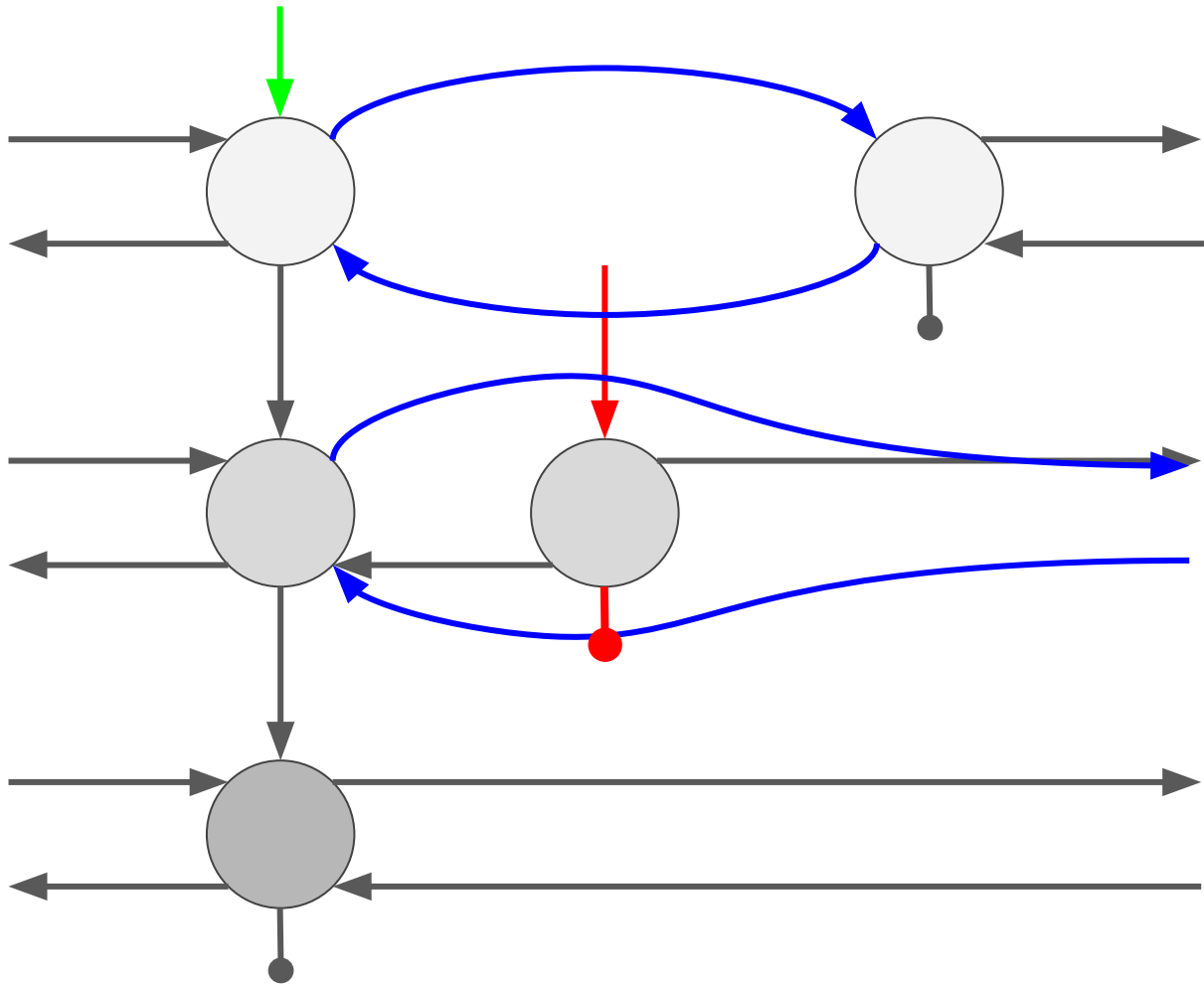
Borrar columna



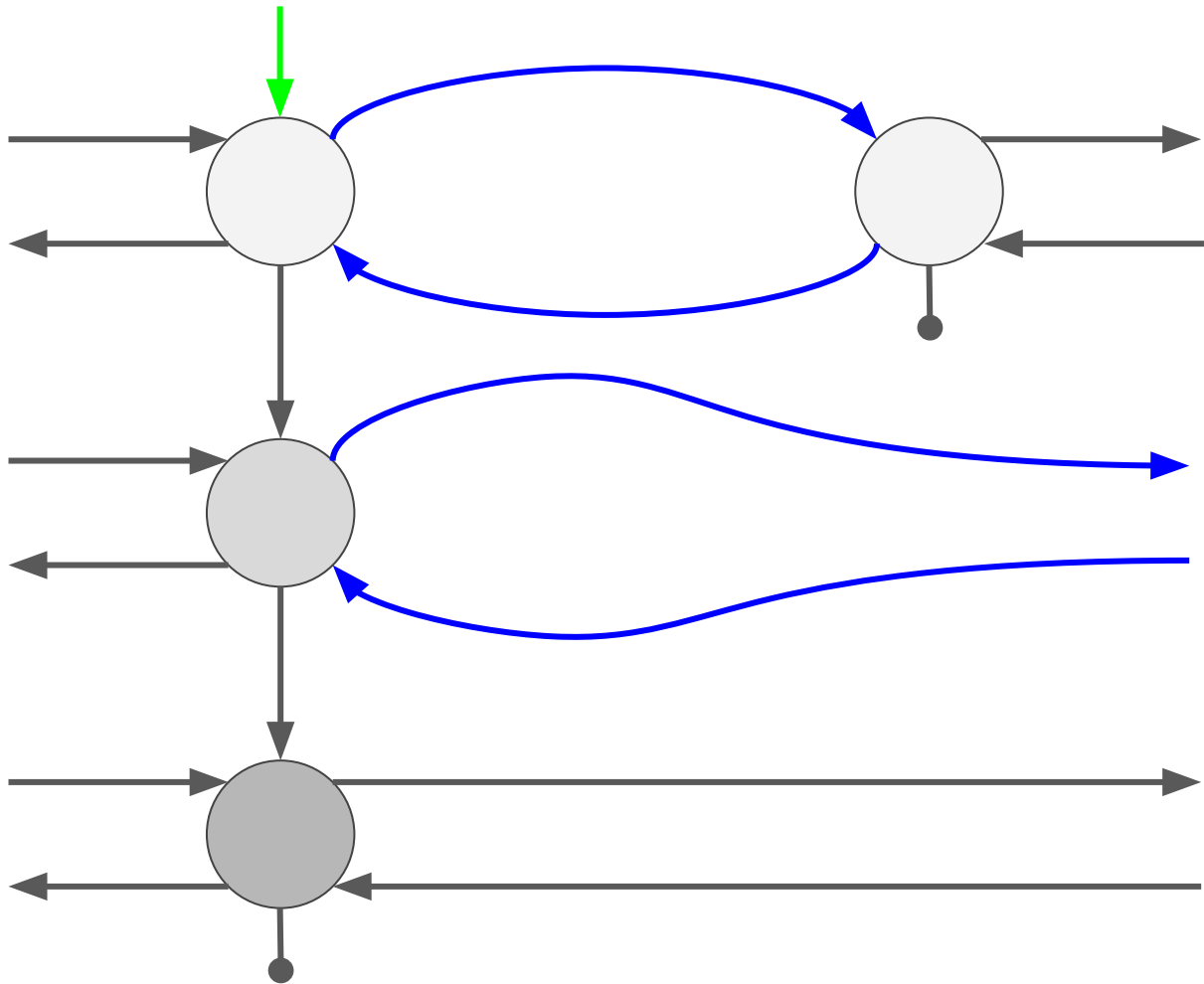
Borrar columna



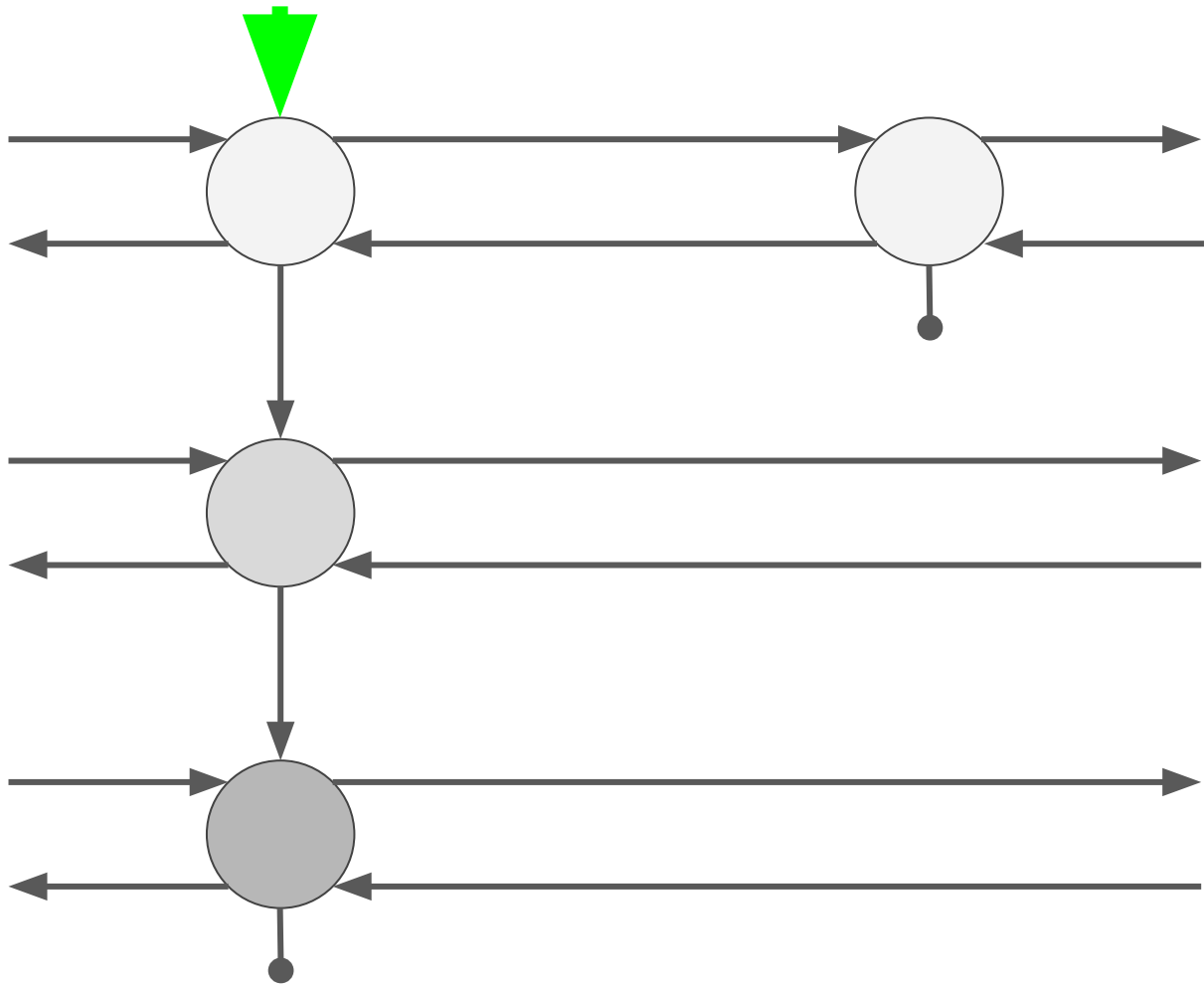
Borrar columna



Borrar columna



Borrar columna



Borrar columna

Setup

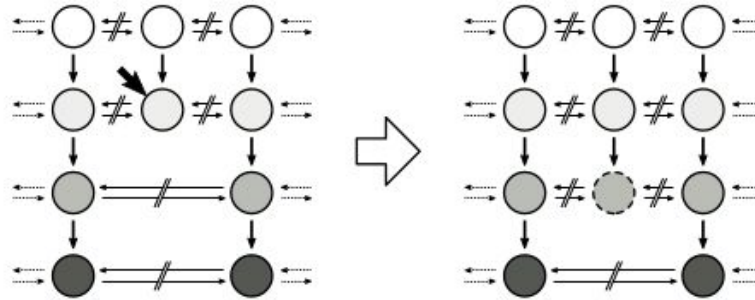
[01] (05, 06, 02)	[02] (01, 07, 03)	[03] (02, 00, 04)	[04] (03, 09, 05)	[05] (04, 00, 01)
[06] (09, 11, 07)	[07] (06, 00, 09)		[09] (07, 14, 06)	
[11] (14, 00, 14)			[14] (11, 00, 11)	

Borrado

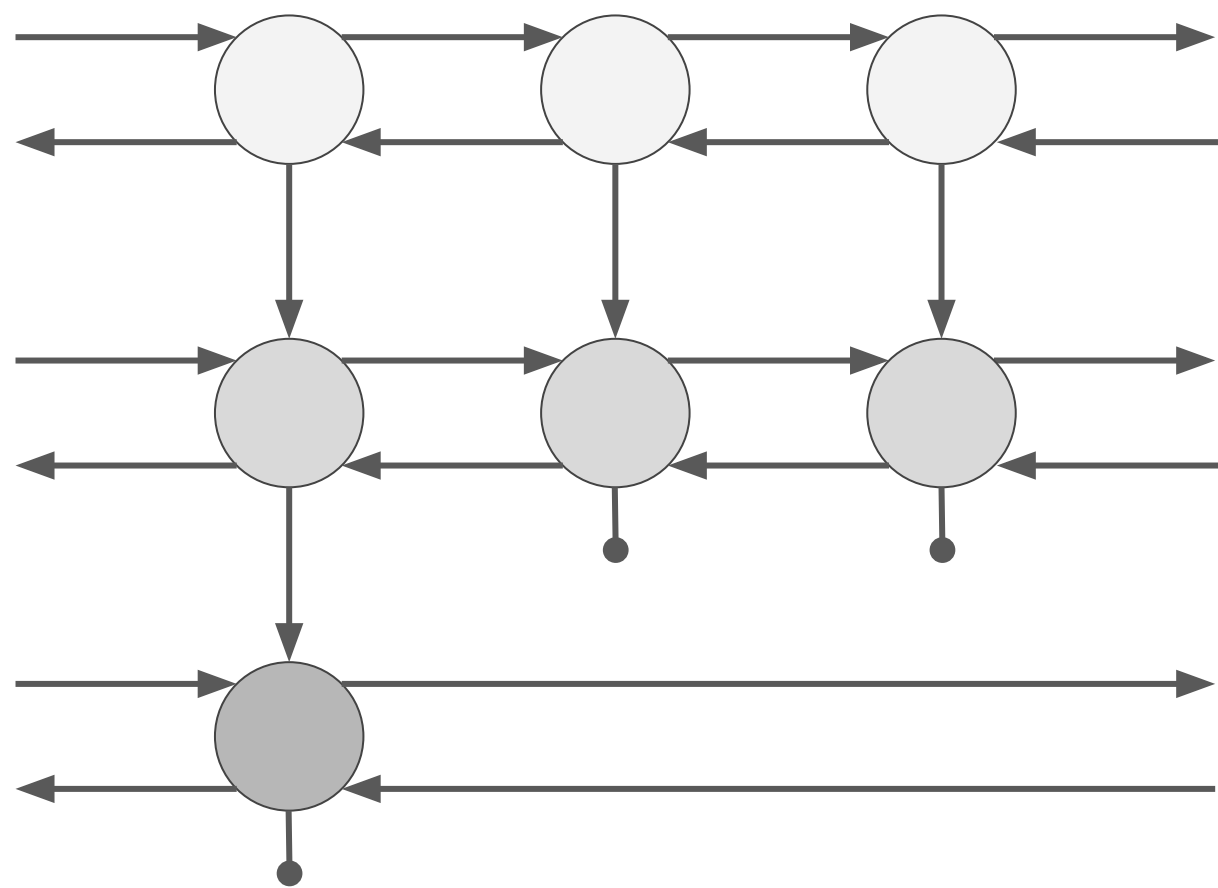
[01] (05, 06, 02)	[02] (01, 07, 03)	[03] (02, 00, 05)	[05] (03, 00, 01)	[01] (05, 06, 02)
[06] (07, 11, 07)	[07] (06, 00, 06)			[06] (07, 11, 07)
[11] (11, 00, 11)				[11] (11, 00, 11)

Agregar abajo

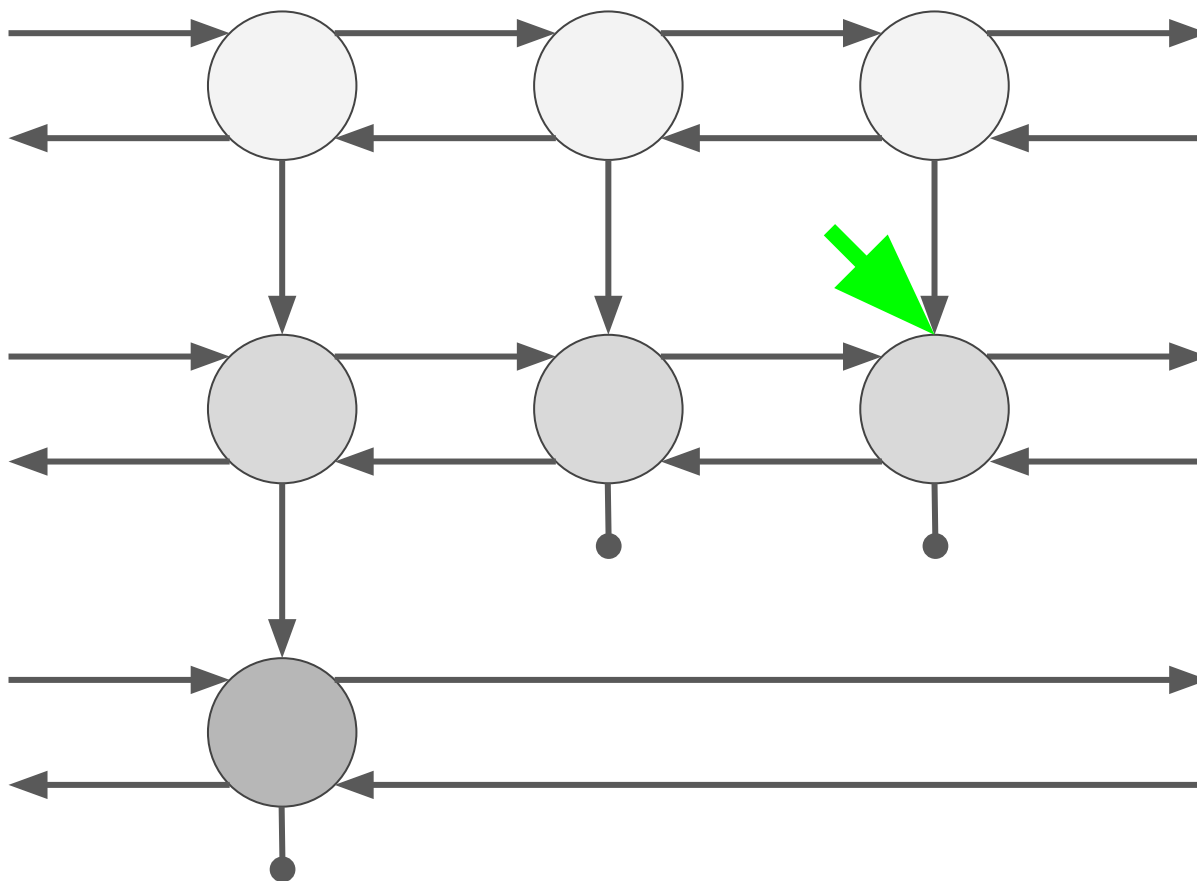
- b. `void agregar_abajo(supernode** sn, int d)` Agrega un nuevo nodo a la lista inmediata inferior del nodo apuntado. Considerar que el nodo donde agregar puede no tener vecinos inmediatos en la lista inferior.



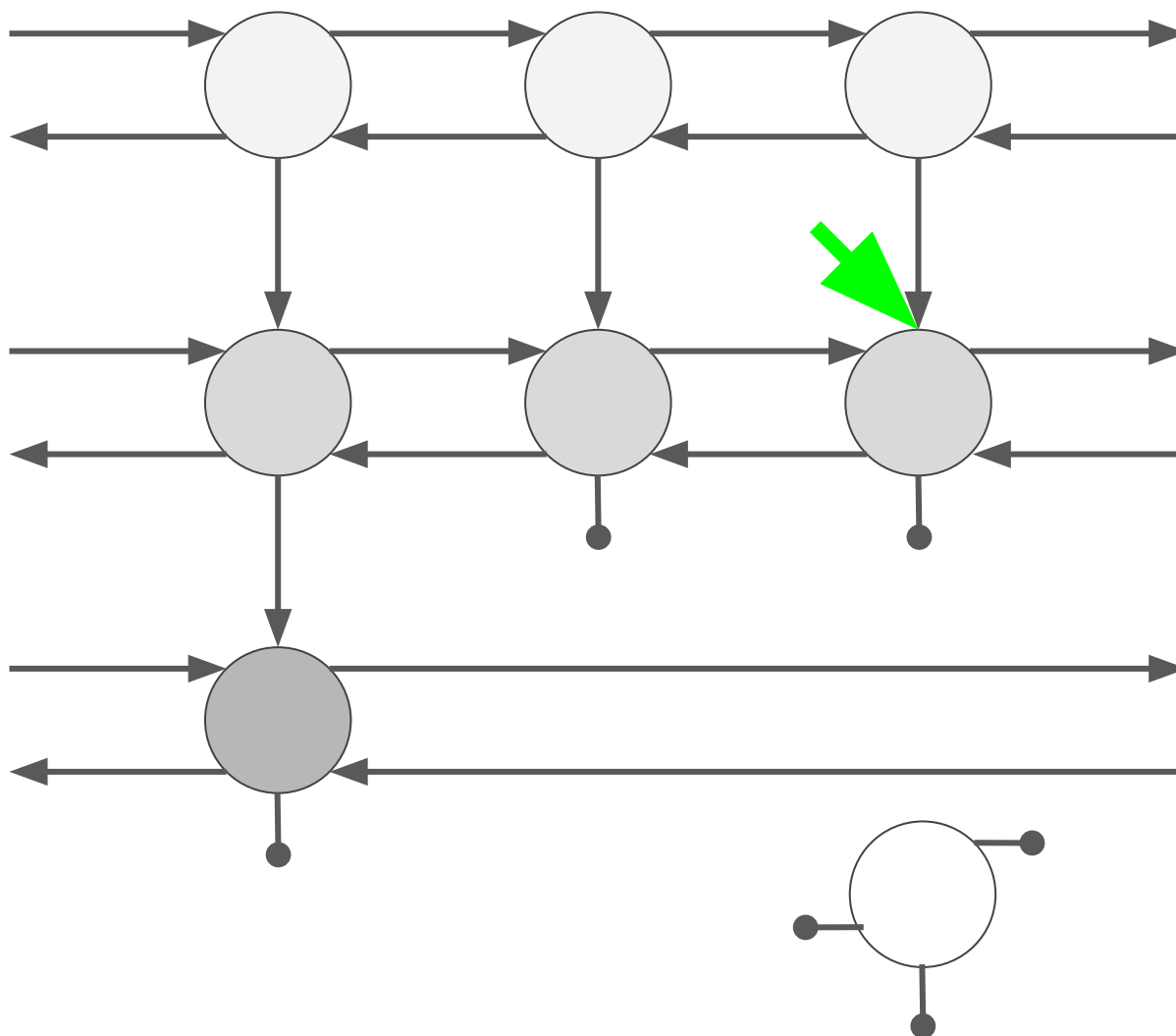
Agregar abajo



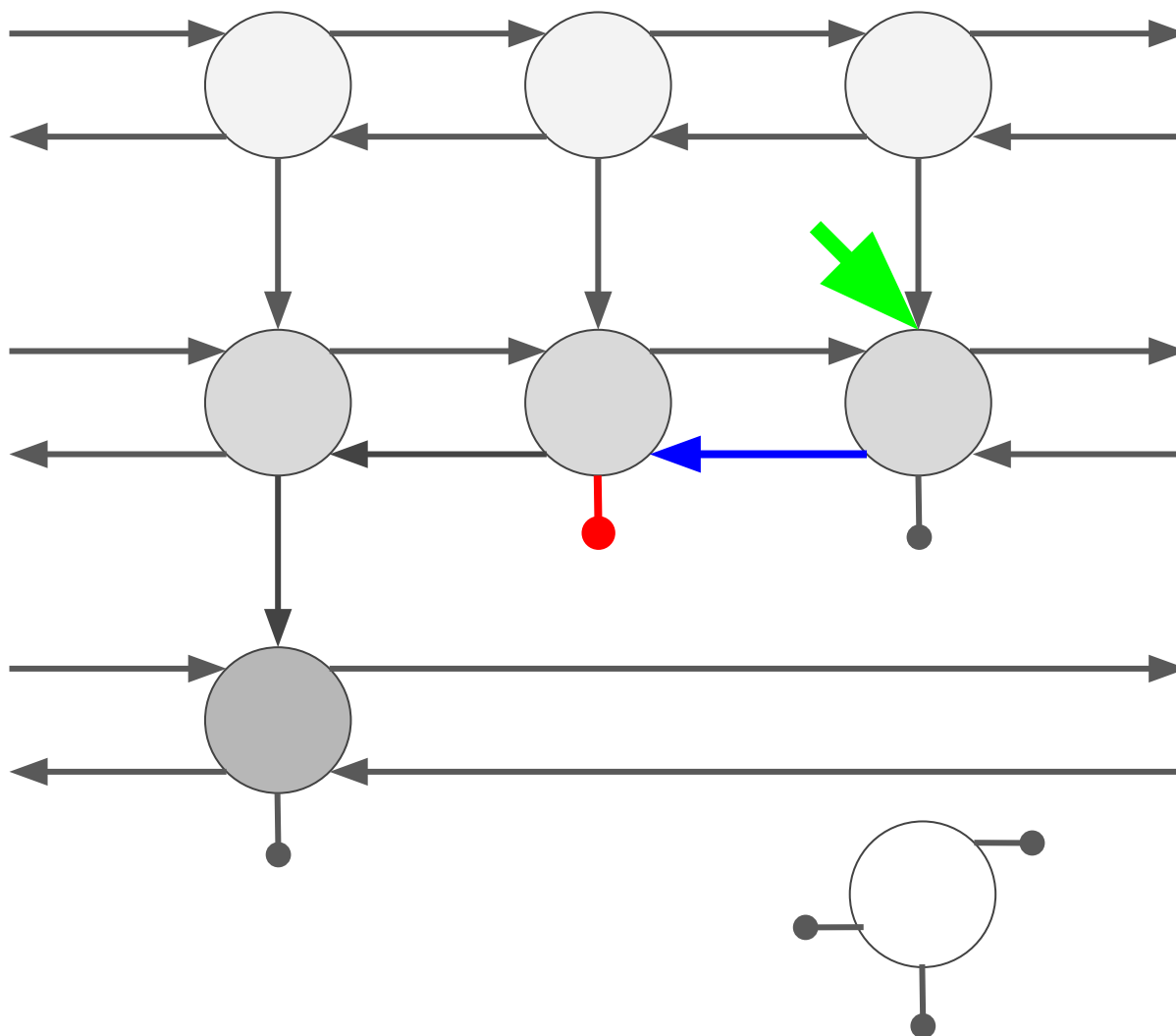
Agregar abajo



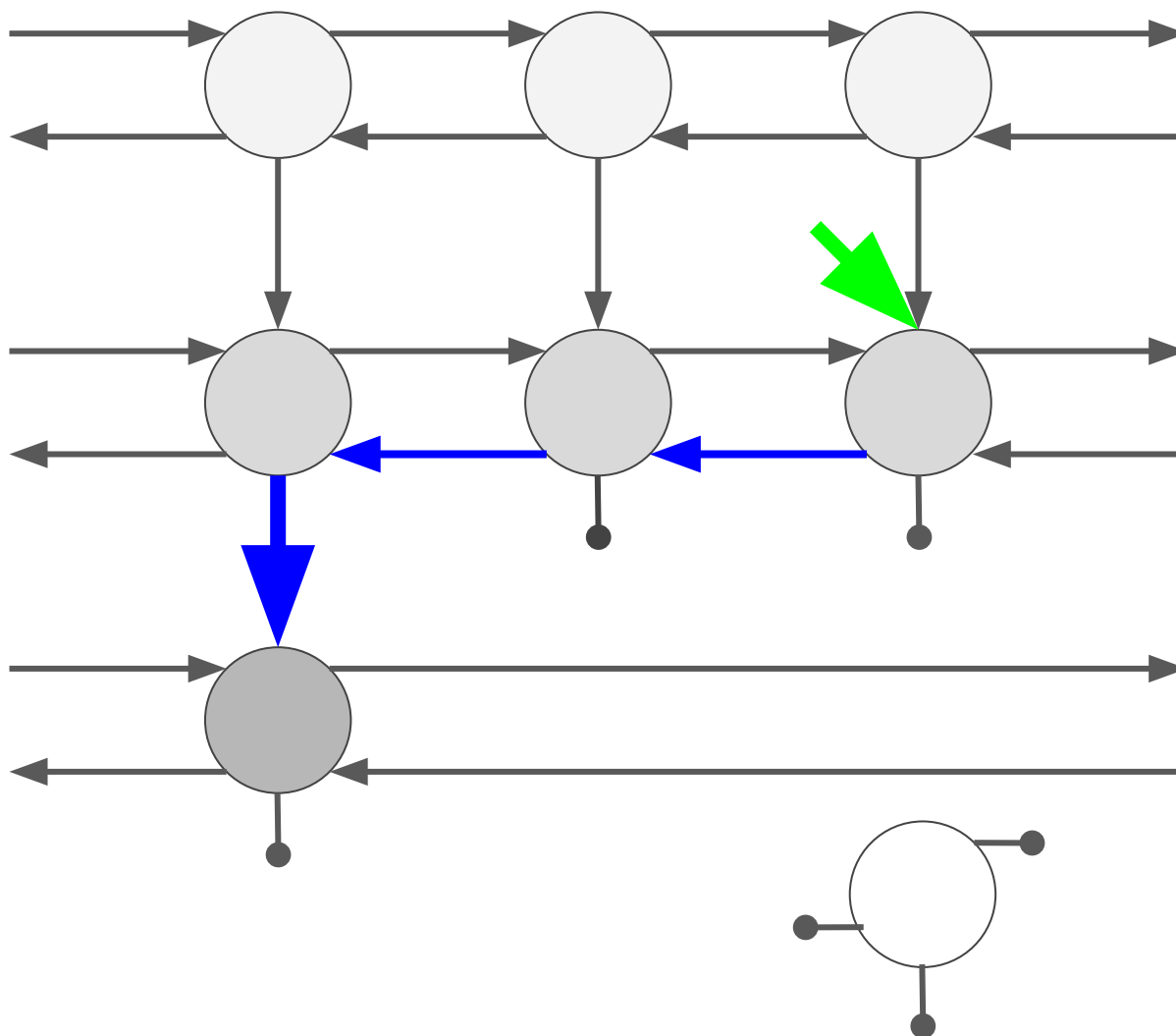
Agregar abajo



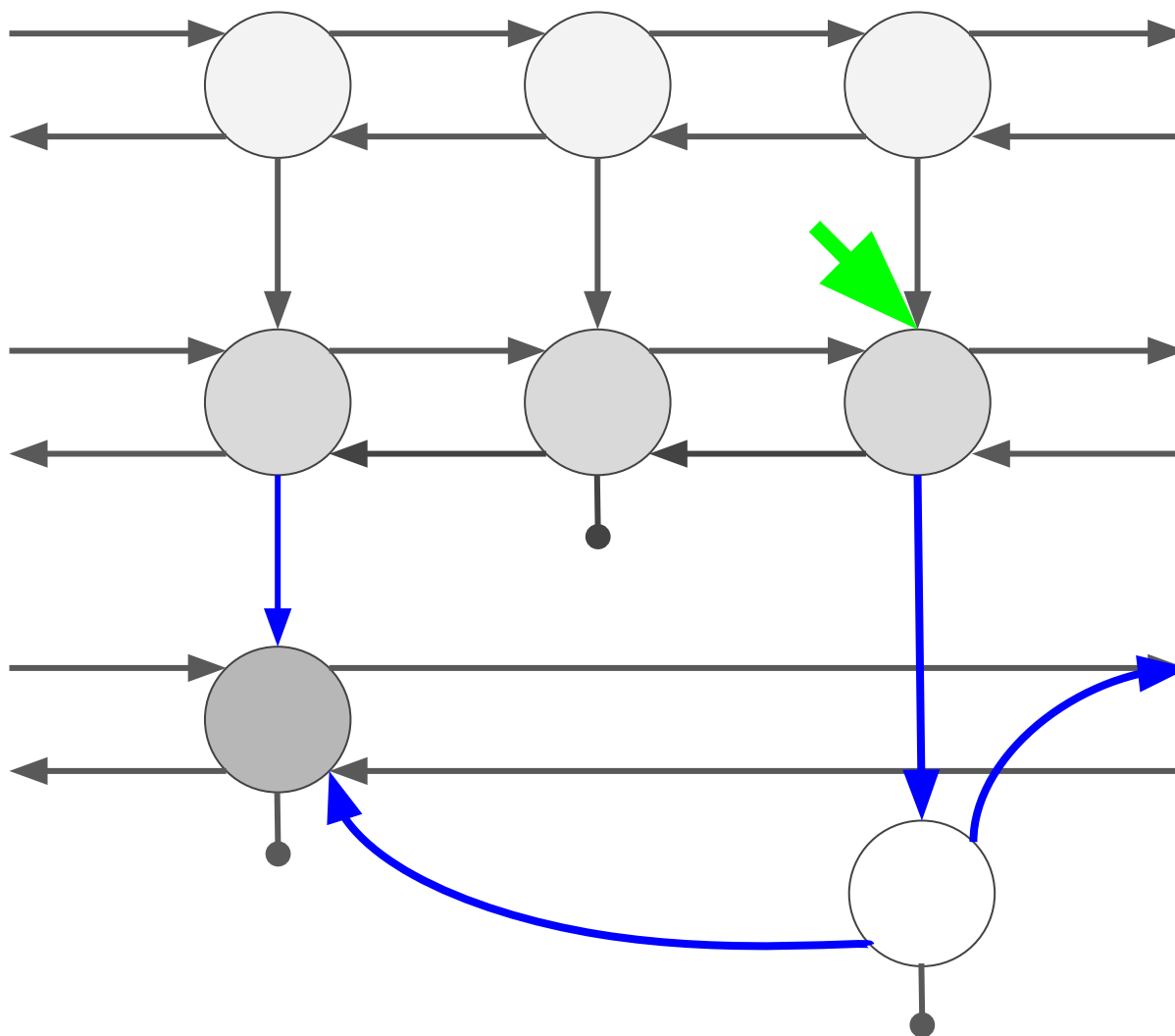
Agregar abajo



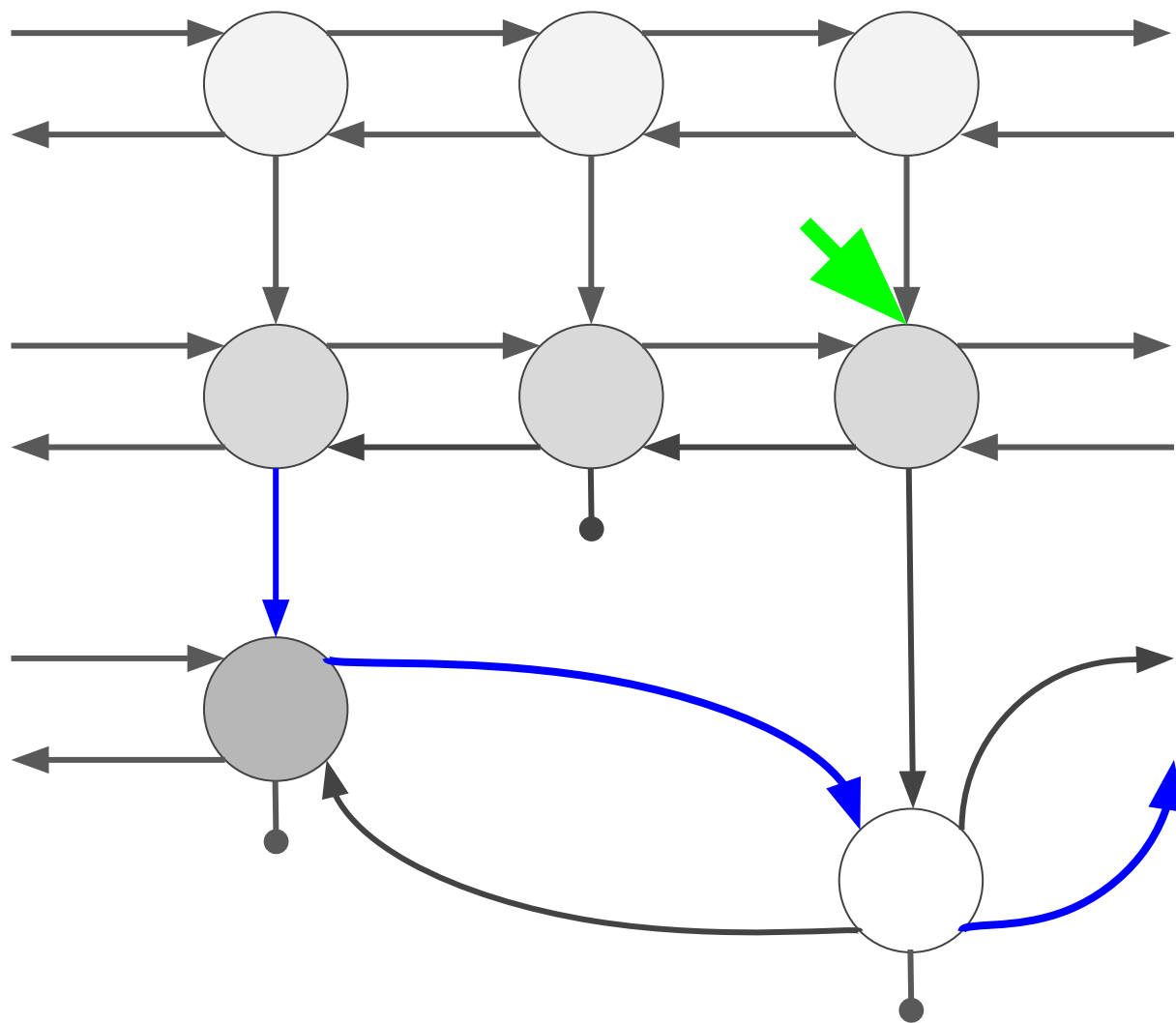
Agregar abajo



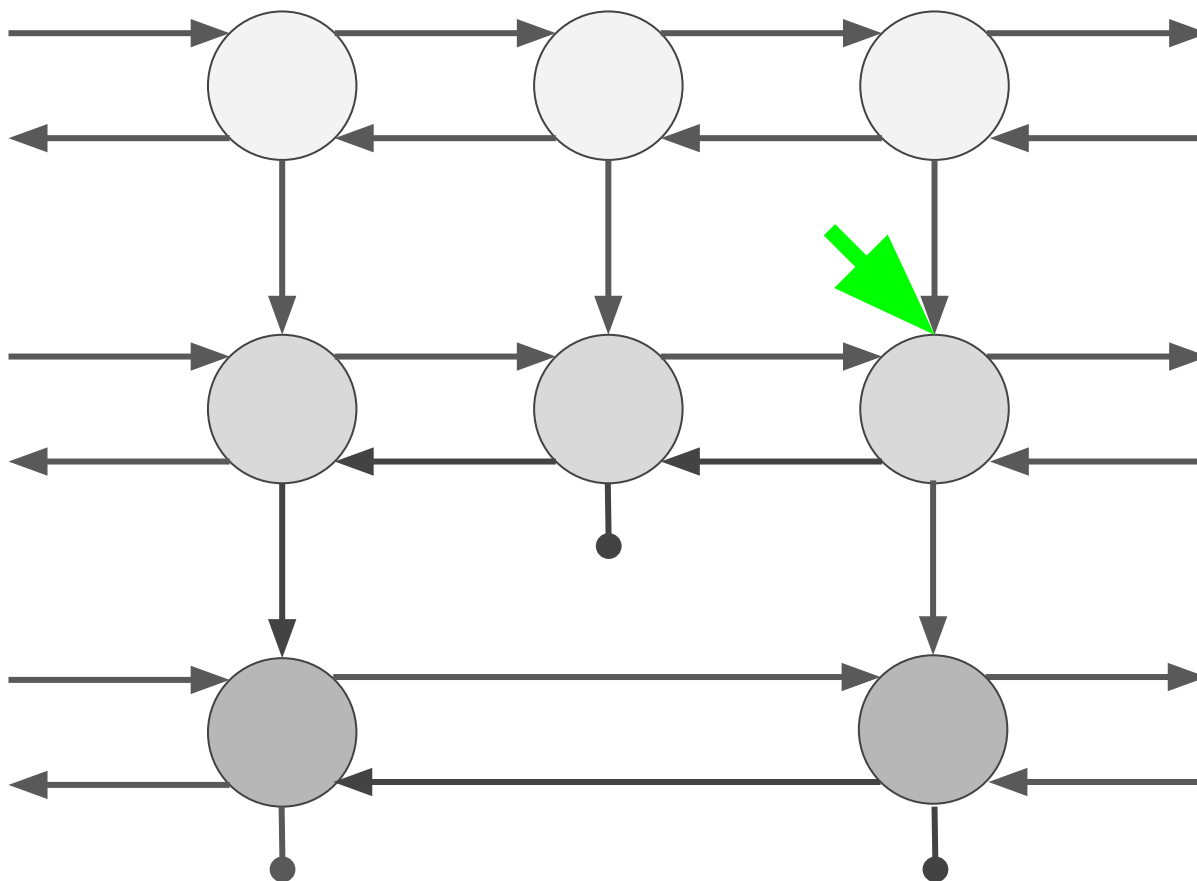
Agregar abajo



Agregar abajo



Agregar abajo



Agregar abajo

Setup

[01] (05, 06, 02)	[02] (01, 07, 03)	[03] (02, 00, 05)	[05] (03, 00, 01)	[01] (05, 06, 02)
[06] (07, 11, 07)	[07] (06, 00, 06)			[06] (07, 11, 07)
[11] (11, 00, 11)				[11] (11, 00, 11)

Agregado

[01] (05, 06, 02)	[02] (01, 07, 03)	[03] (02, 00, 05)	[05] (03, 16, 01)	[01] (05, 06, 02)
[06] (16, 11, 07)	[07] (06, 00, 16)		[16] (07, 00, 06)	[06] (16, 11, 07)
[11] (11, 00, 11)				[11] (11, 00, 11)