



Breadth First Search

Abdullah Mughrabi

History

“BFS and its application in finding connected components of graphs were invented in 1945 by Konrad Zuse and Michael Burke, in their (*rejected*) Ph.D. thesis on the Plankalkül programming language, but this was not published until 1972.”

Breadth First Search (or Traversal) BFS

- Is an algorithm for *traversing* or *searching* tree or graph data structure.
- A *tree* is a special kind of graph that does not contain cycles. (*if we start at a random node we can't come to the same node again.*)

It is all about Graphs!!!

Basic idea

- We start traversing the nodes of graph in layers (*frontiers*.)
- Starting at a *source* node and keep traversing(searching) until we find a target node (*search*) or explored our entire possible paths (*travers*).
- The *frontier* contains nodes that we have seen but have not explored yet.
- Each iteration we take a node off the frontier, then add its neighbors to the next frontier.

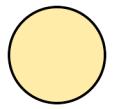
BFS is not a niche

- Shortest Path and Minimum Spanning Tree for unweighted graph.
- Peer to Peer Networks.
- Crawlers in Search Engines.
- GPS Navigation systems.
- Social Networking Websites.
- Broadcasting in Network.
- In Garbage Collection.
- Cycle detection in undirected graph.

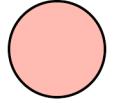
BFS is not a niche

- Ford–Fulkerson algorithm.
- To test if a graph is Bipartite.
- Path Finding.
- Finding all nodes within one connected component.
- Many algorithms like Prim's Minimum Spanning Tree and Dijkstra's Single Source Shortest Path use structure similar to Breadth First Search.

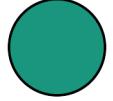
There can be many more applications as Breadth First Search is one of the core algorithms for Graphs.



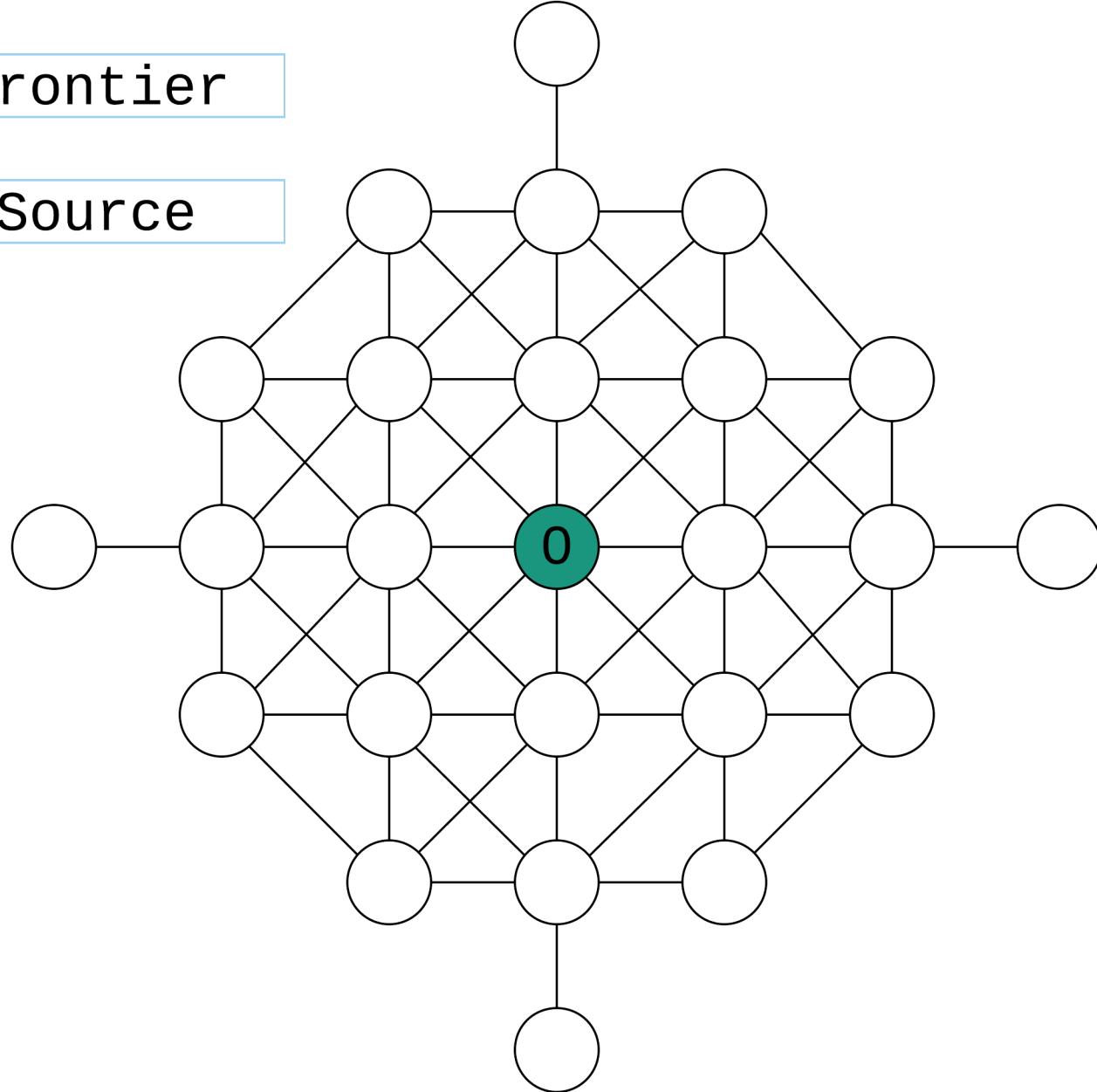
Next

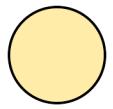


Frontier

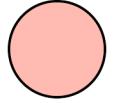


Source

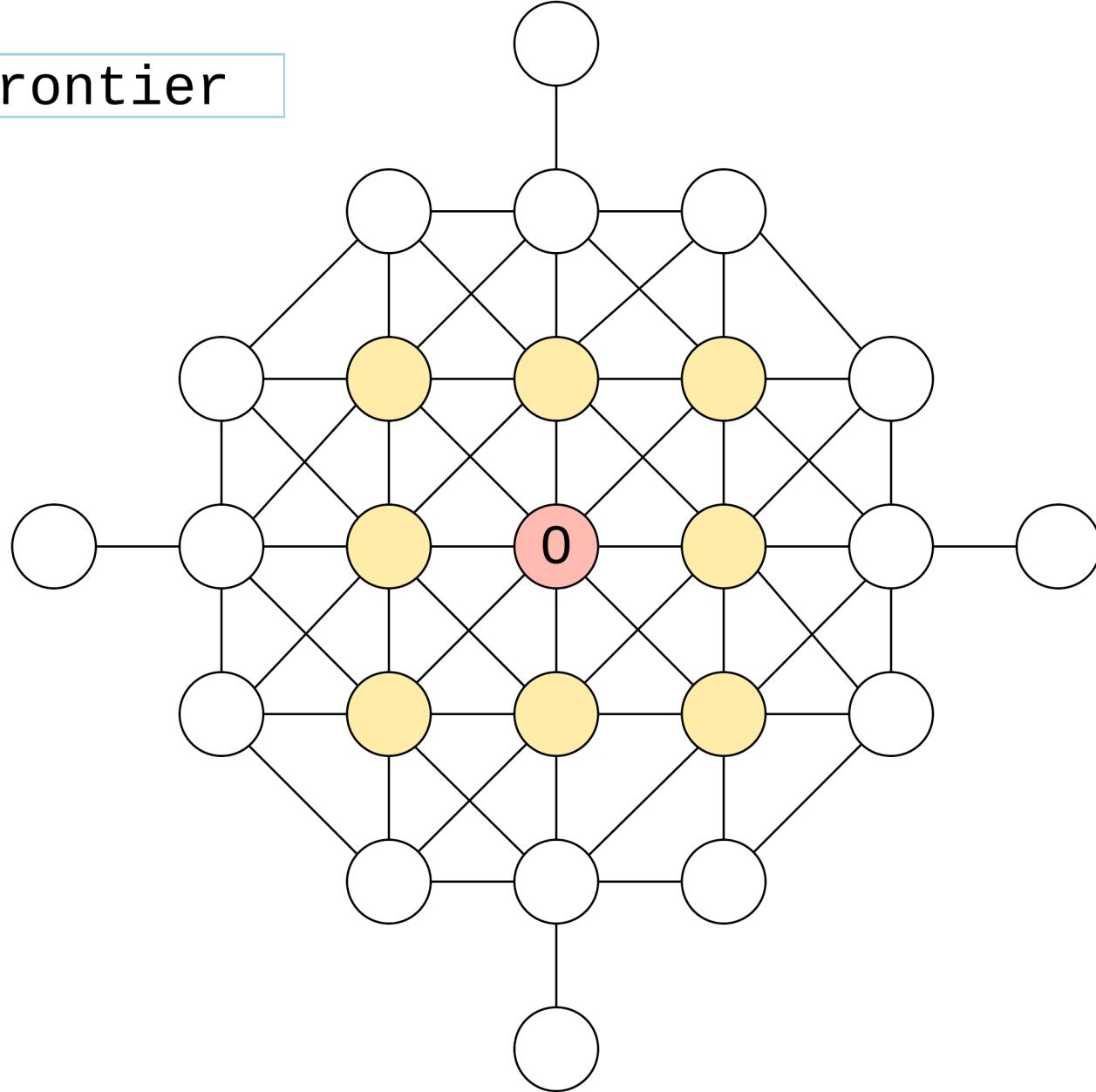


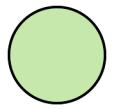


Next

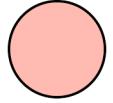


Frontier

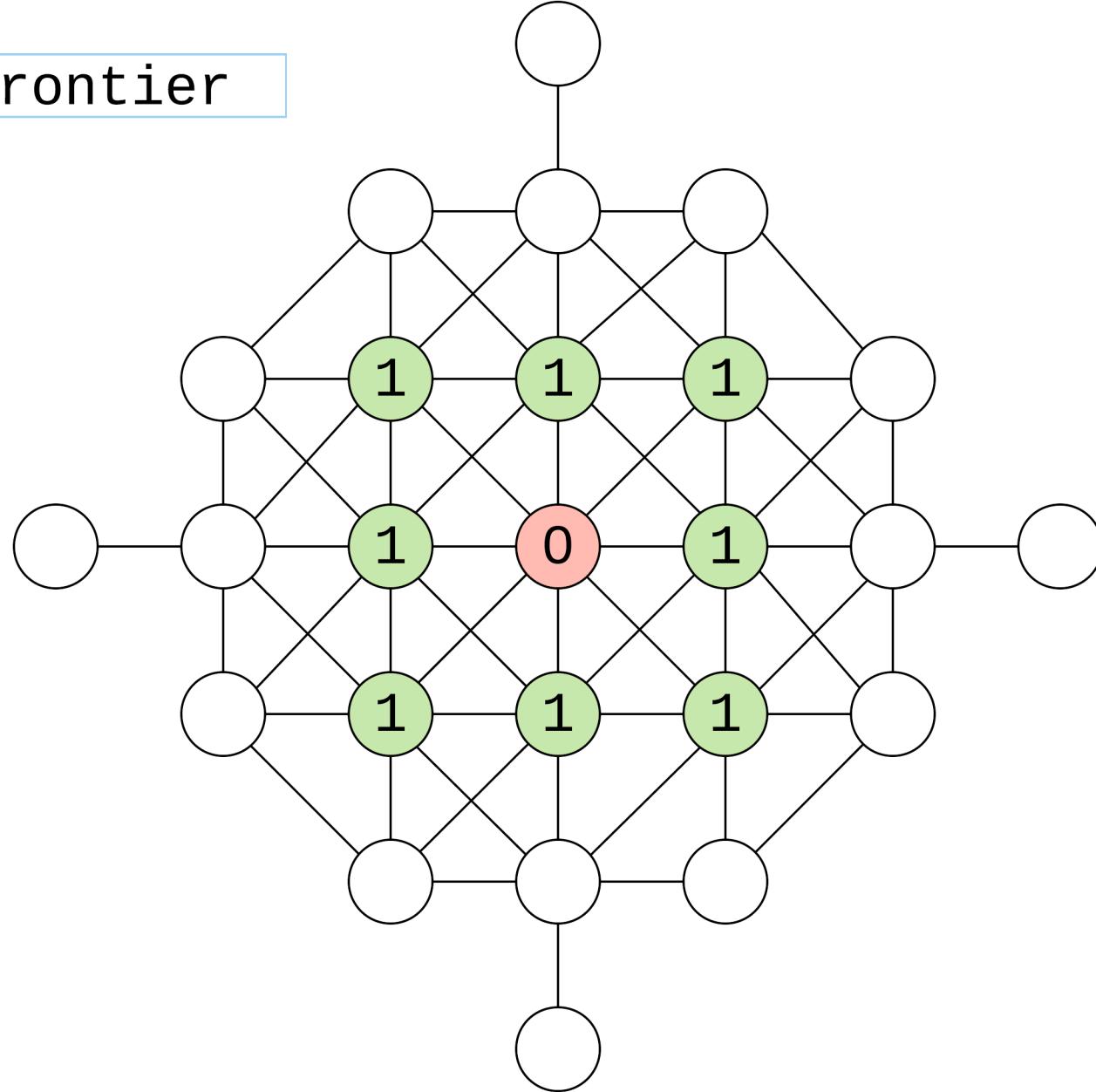


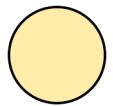


Next

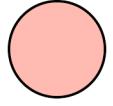


Frontier

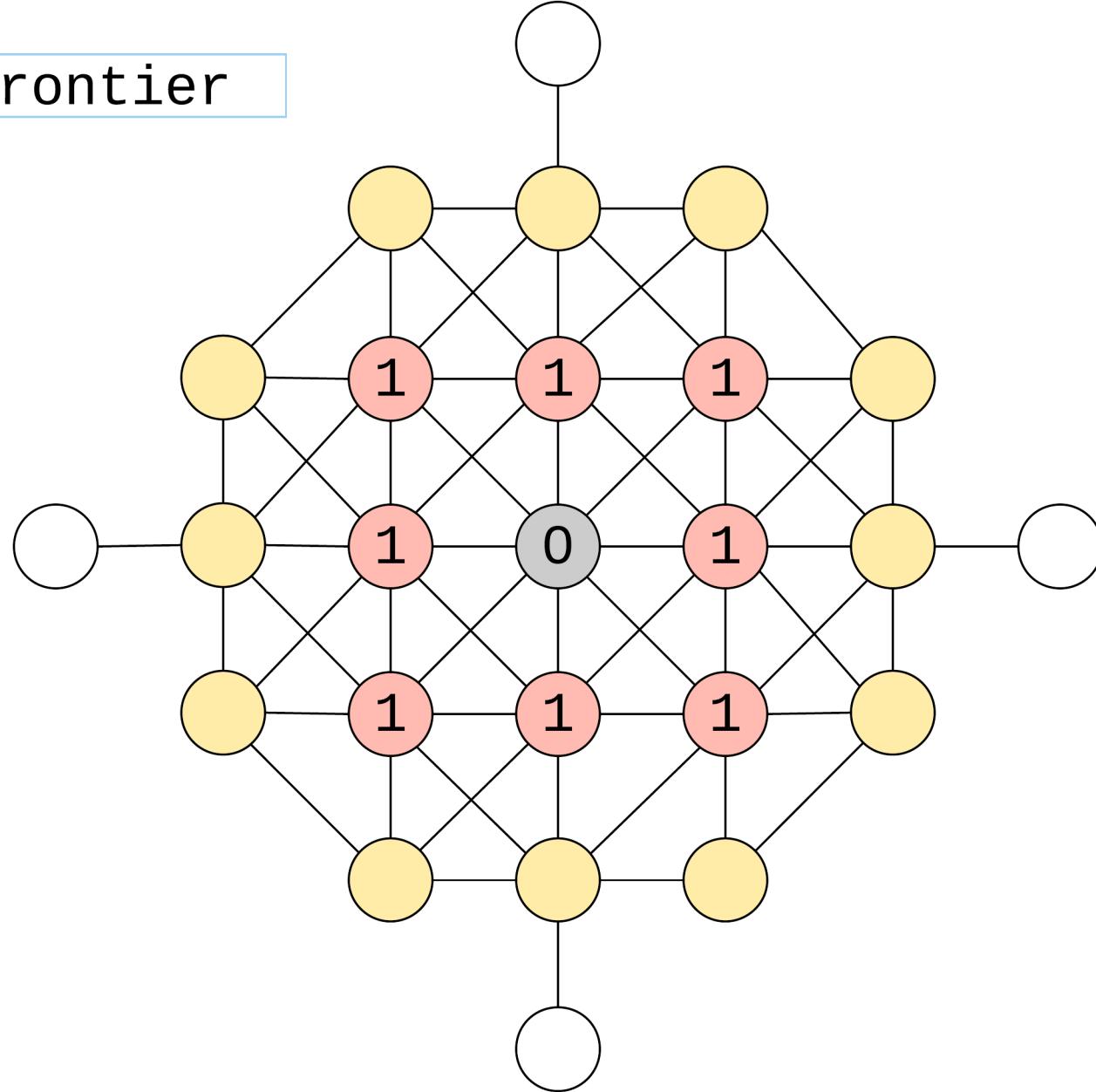


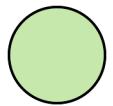


Next

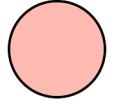


Frontier

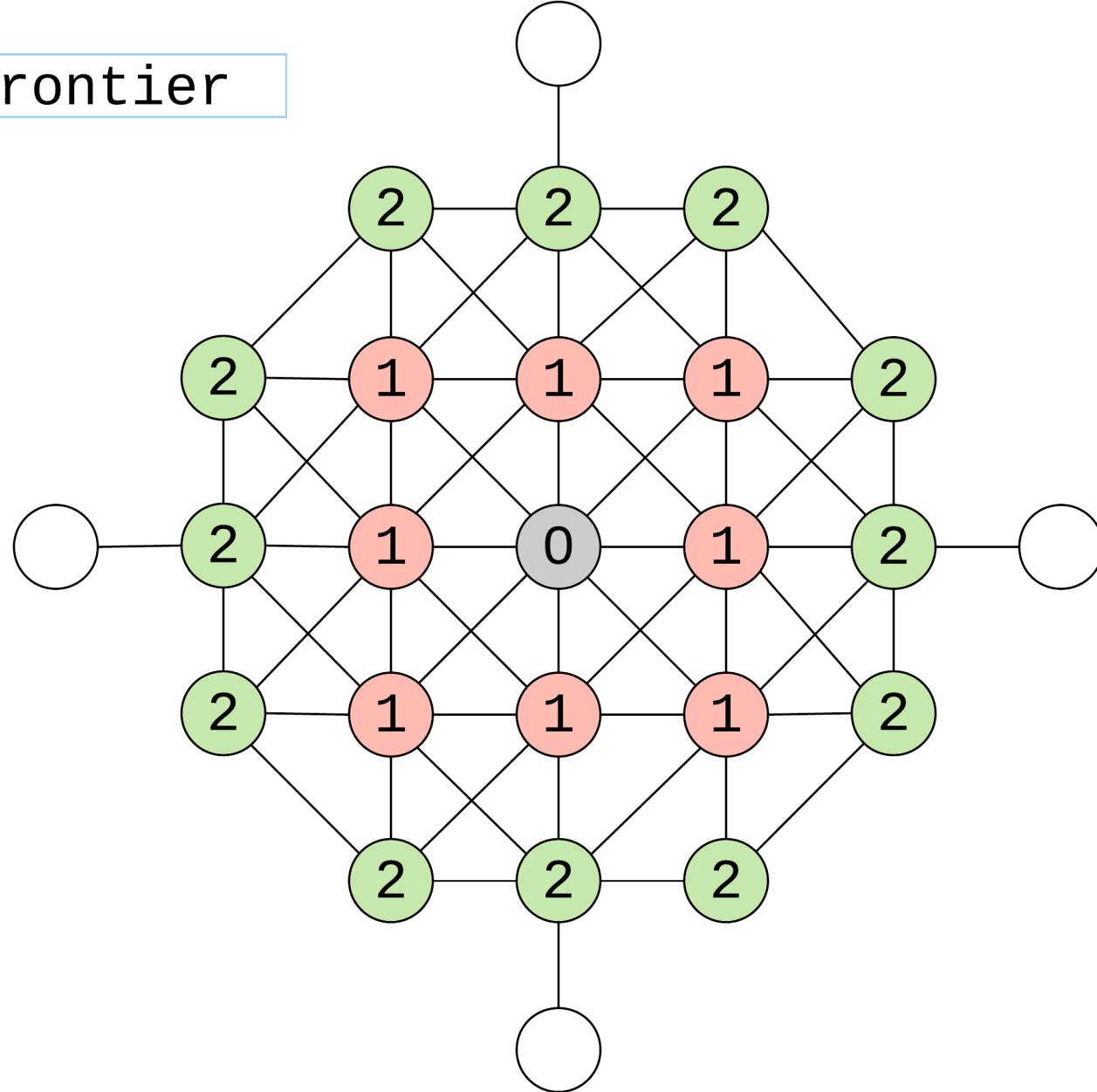


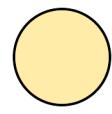


Next

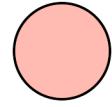


Frontier

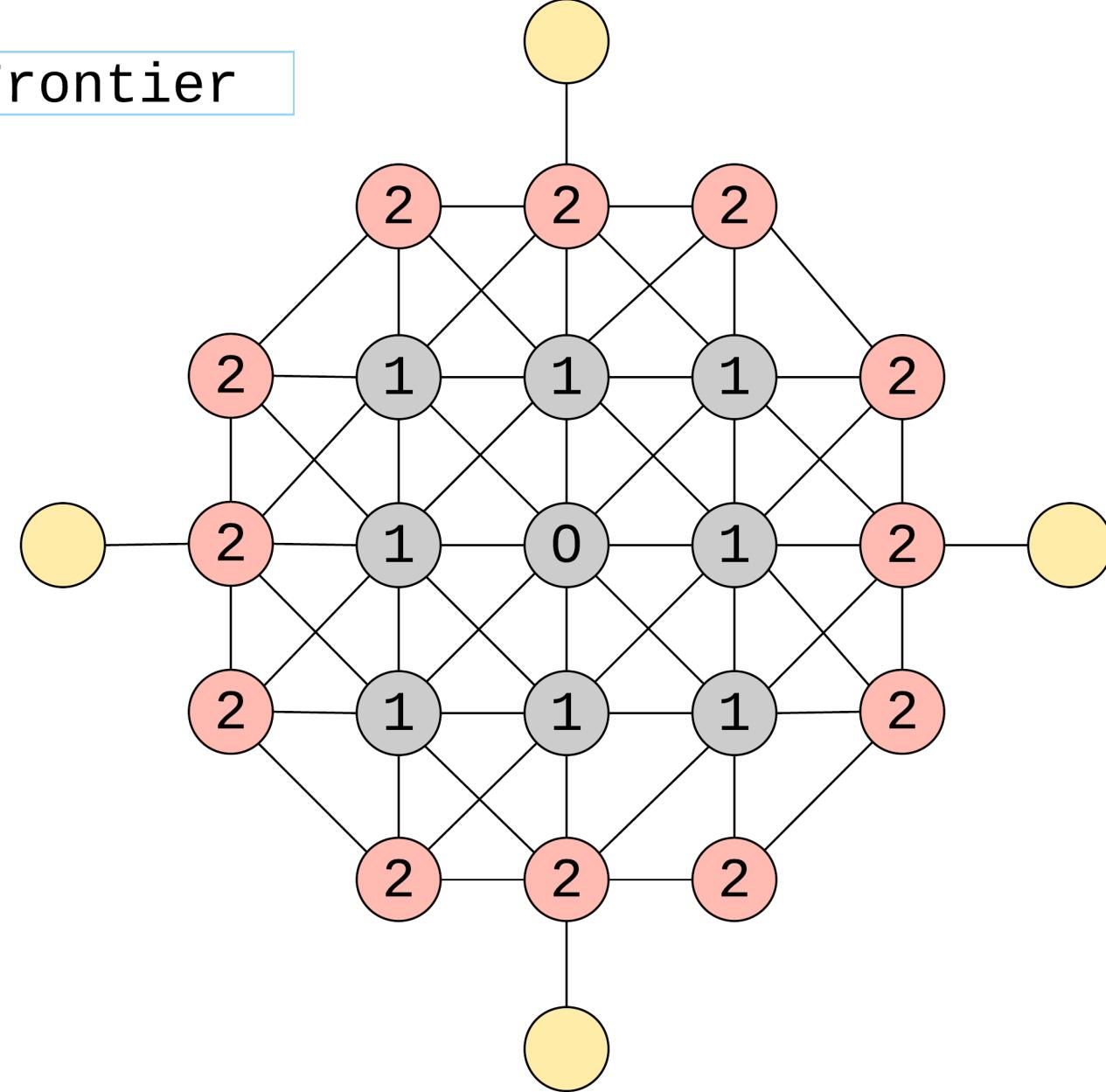


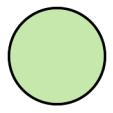


Next

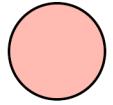


Frontier

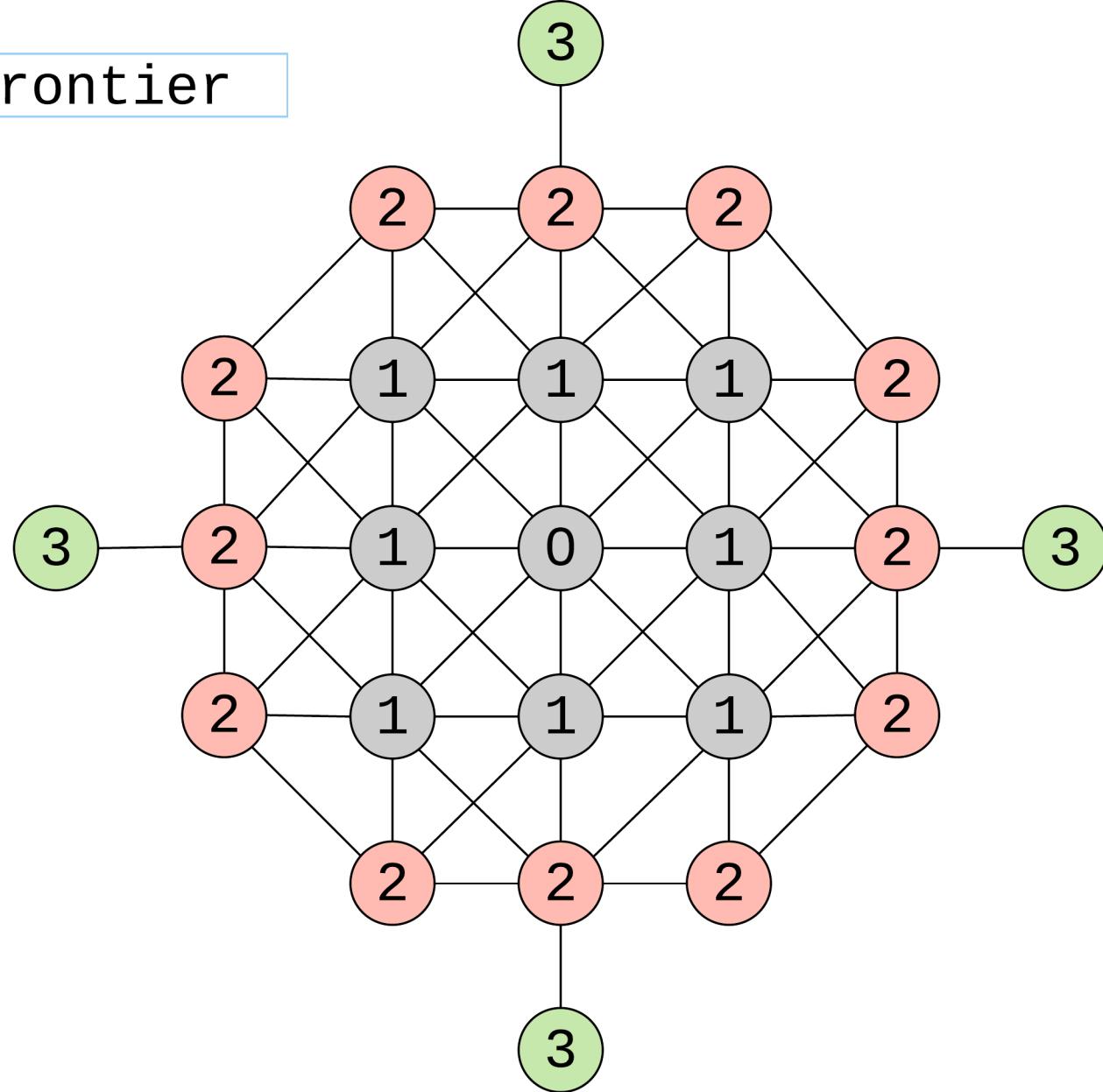


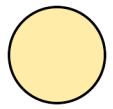


Next

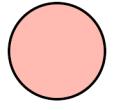


Frontier

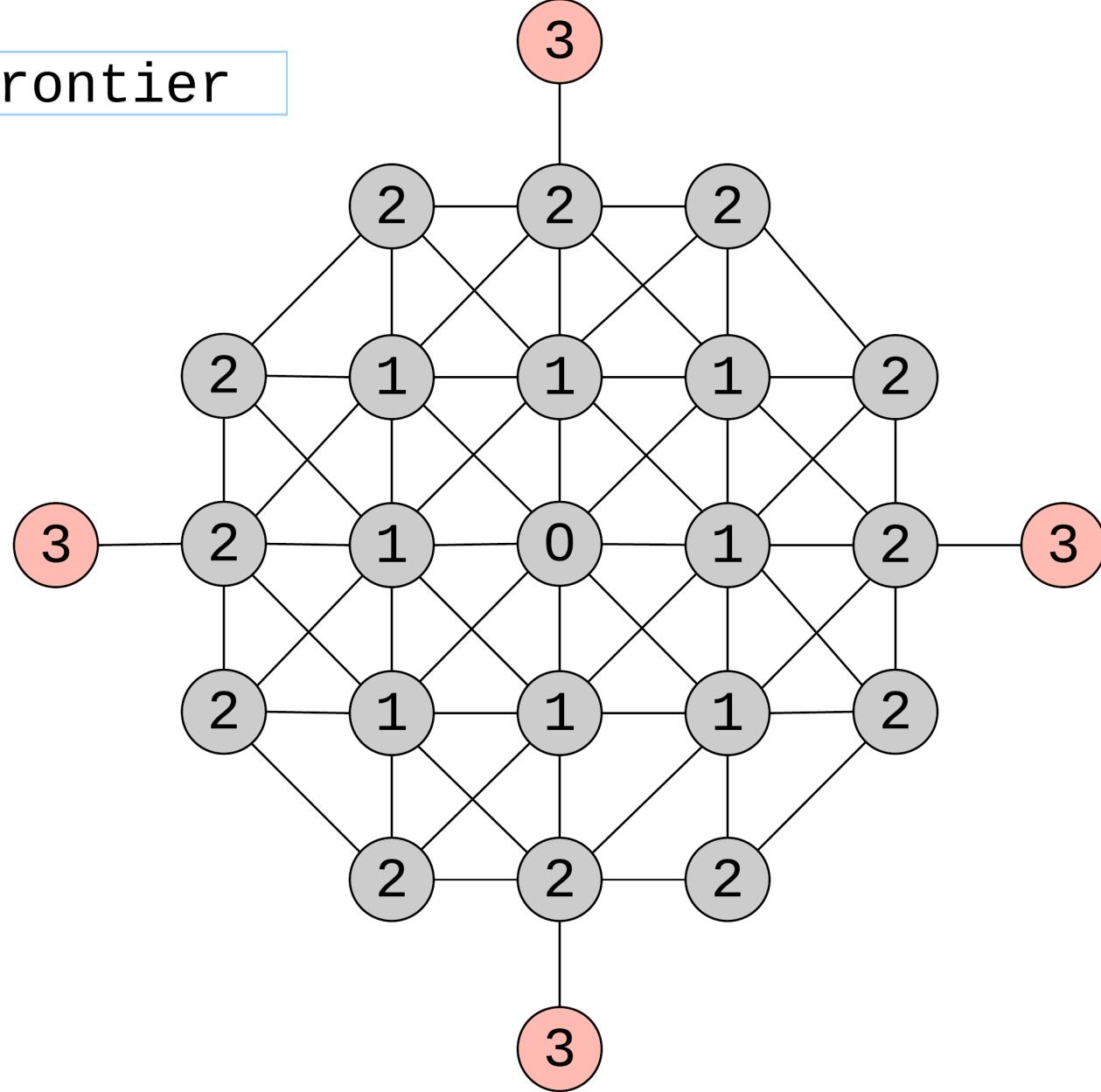


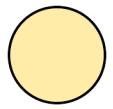


Next

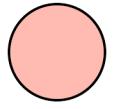


Frontier

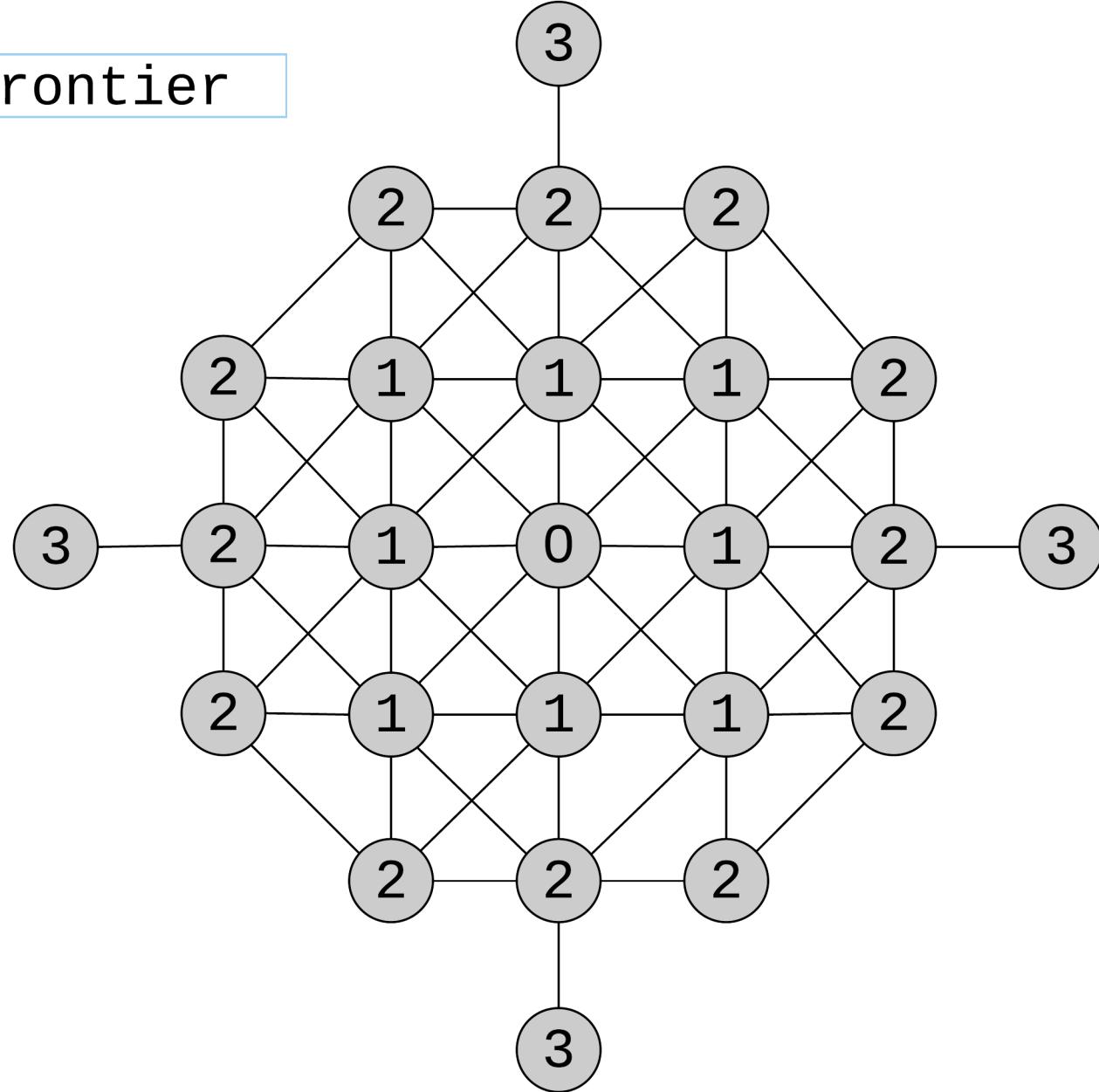


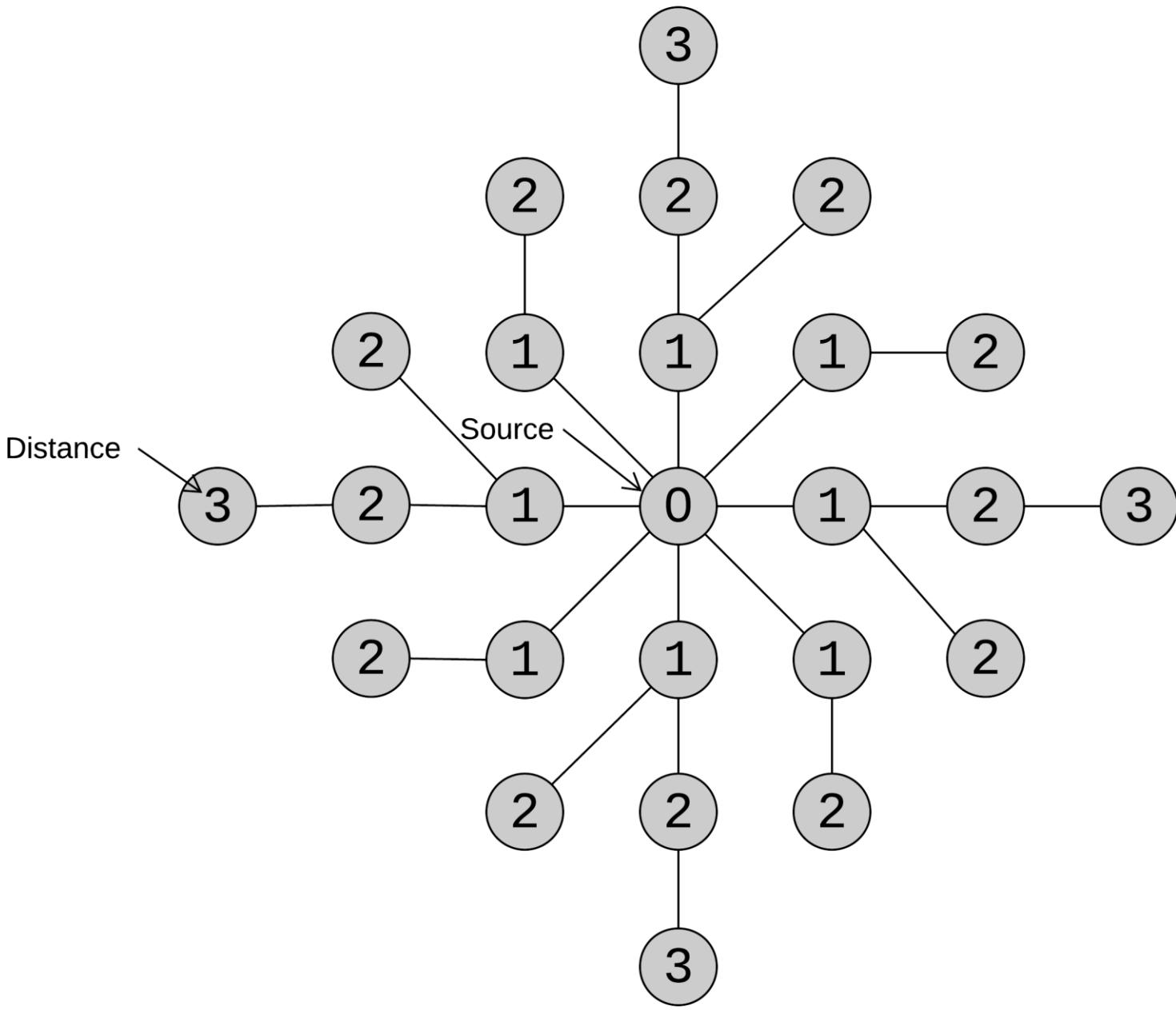


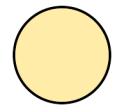
Next



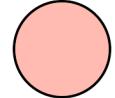
Frontier



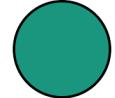




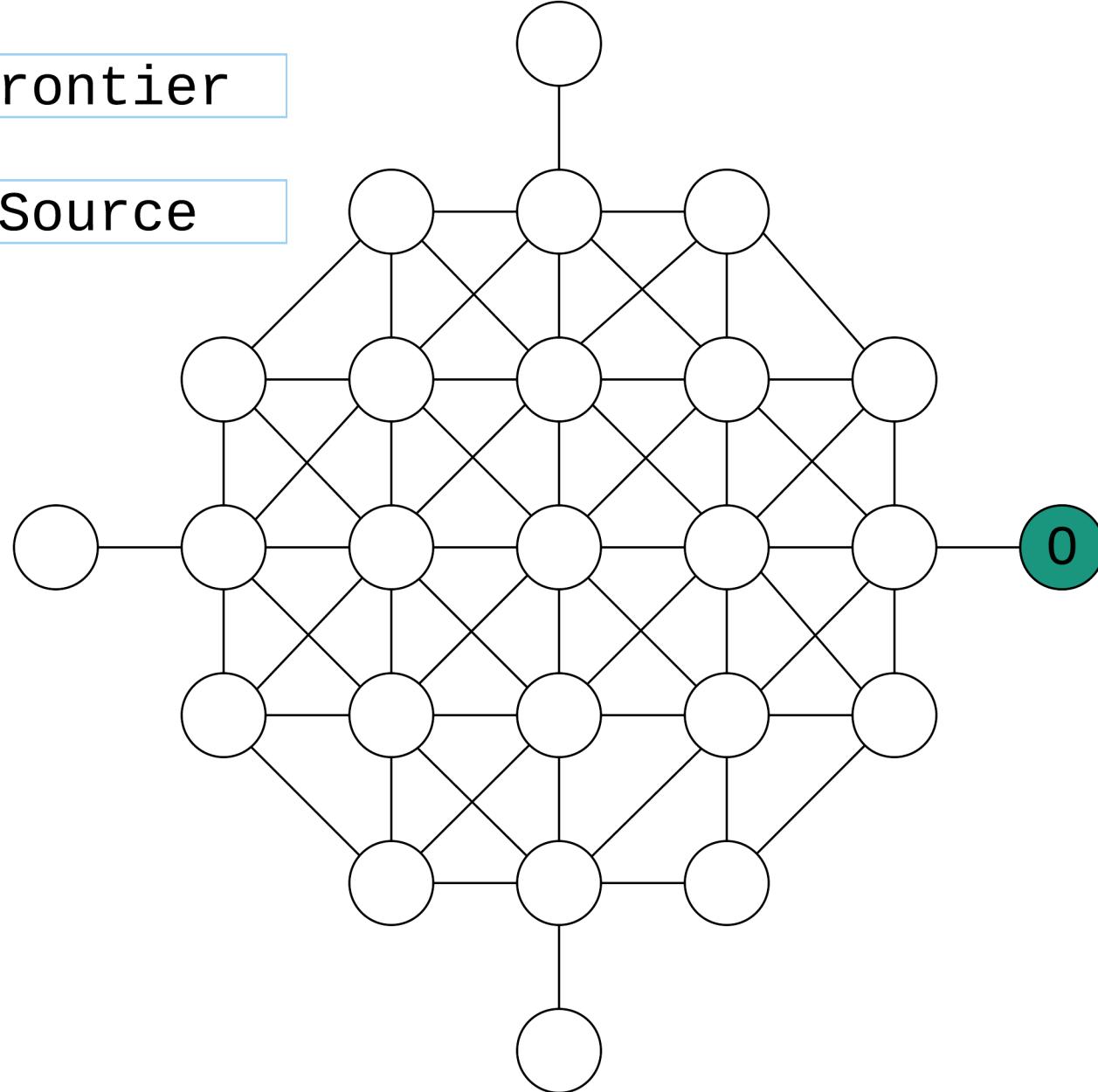
Next

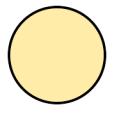


Frontier

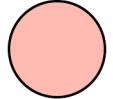


Source

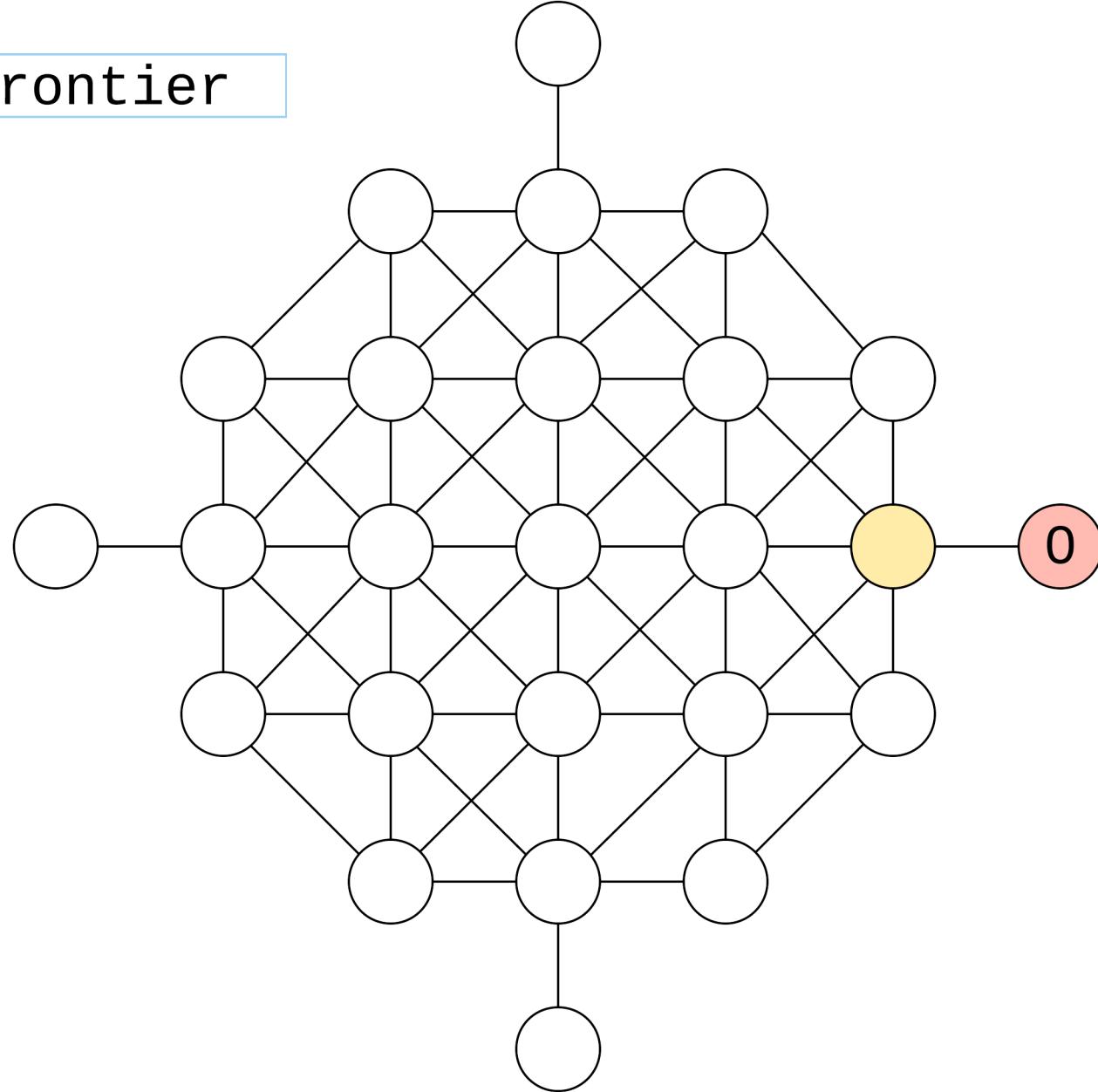


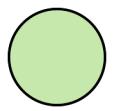


Next

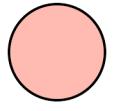


Frontier

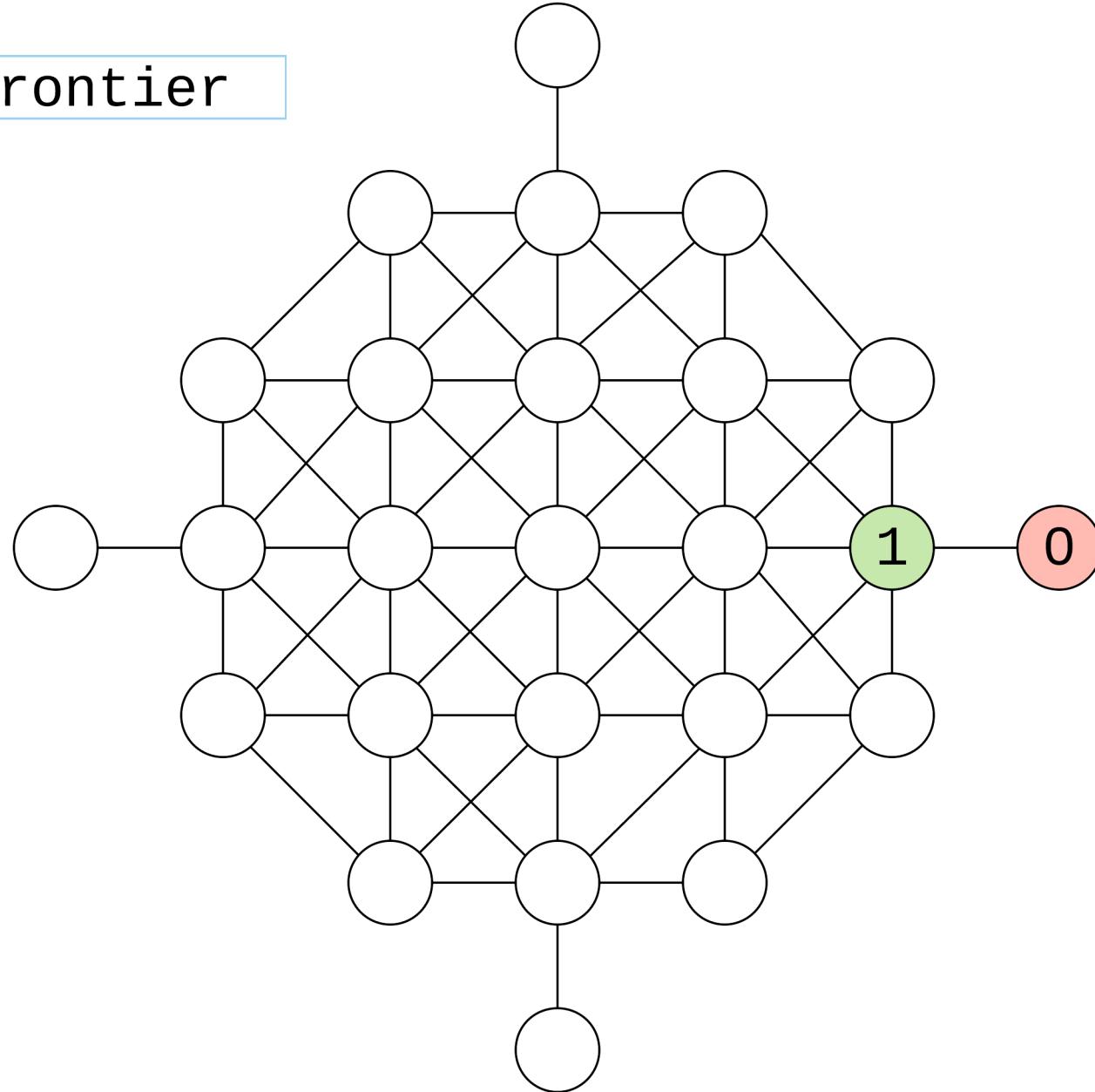


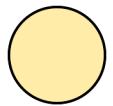


Next

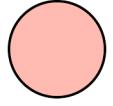


Frontier

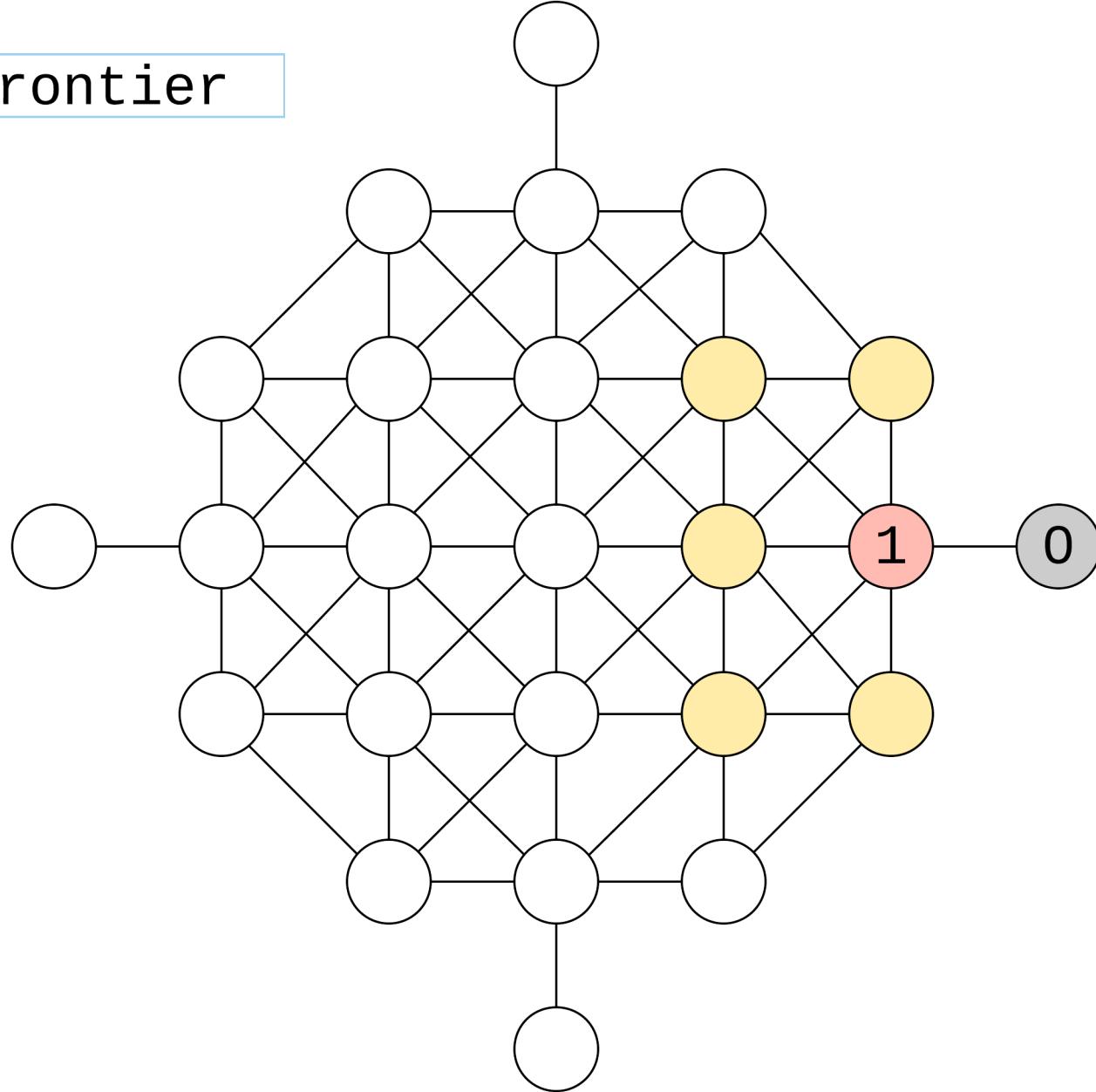


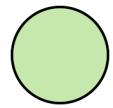


Next

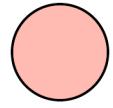


Frontier

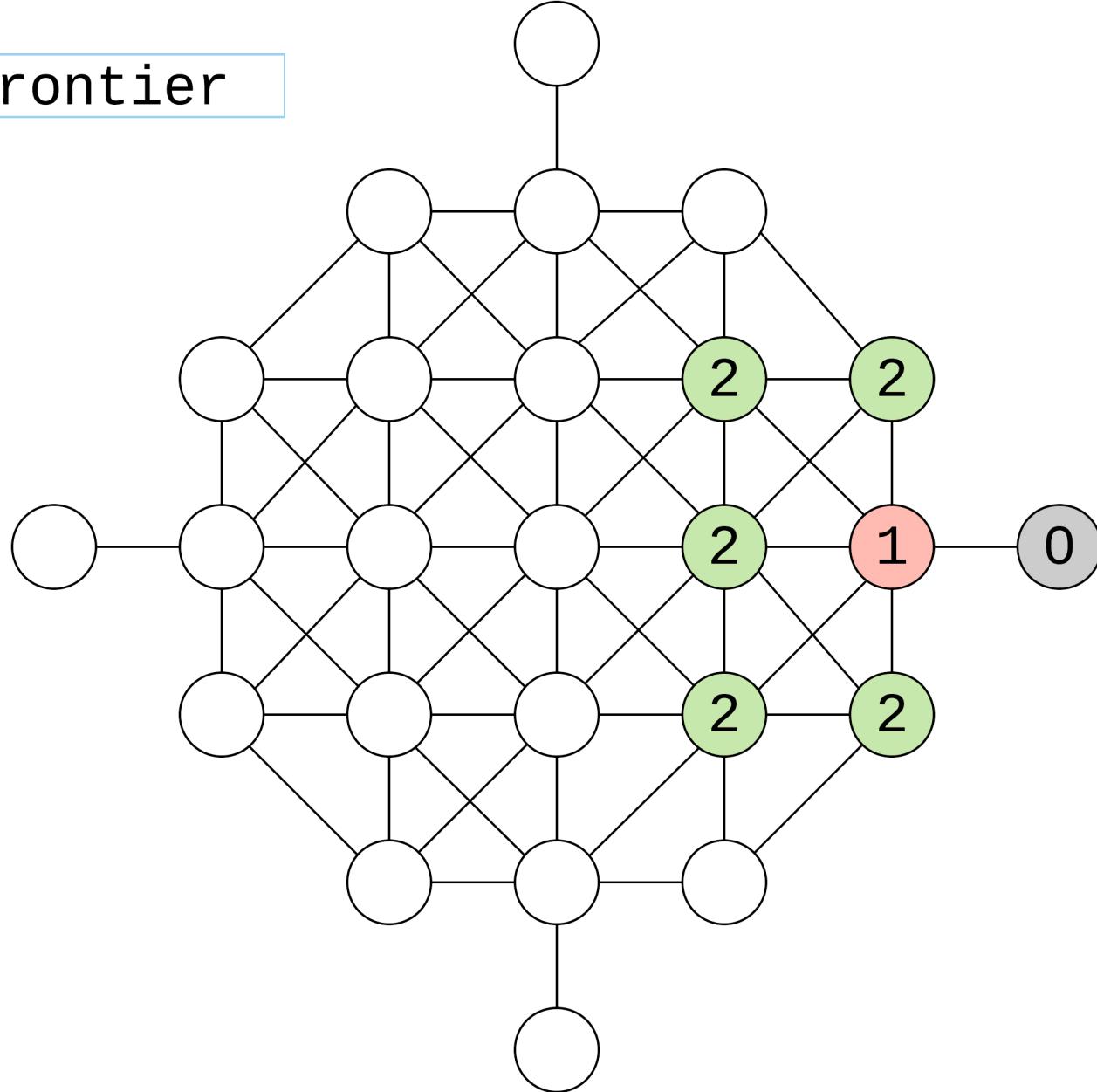


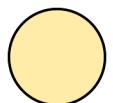


Next

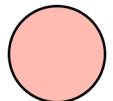


Frontier

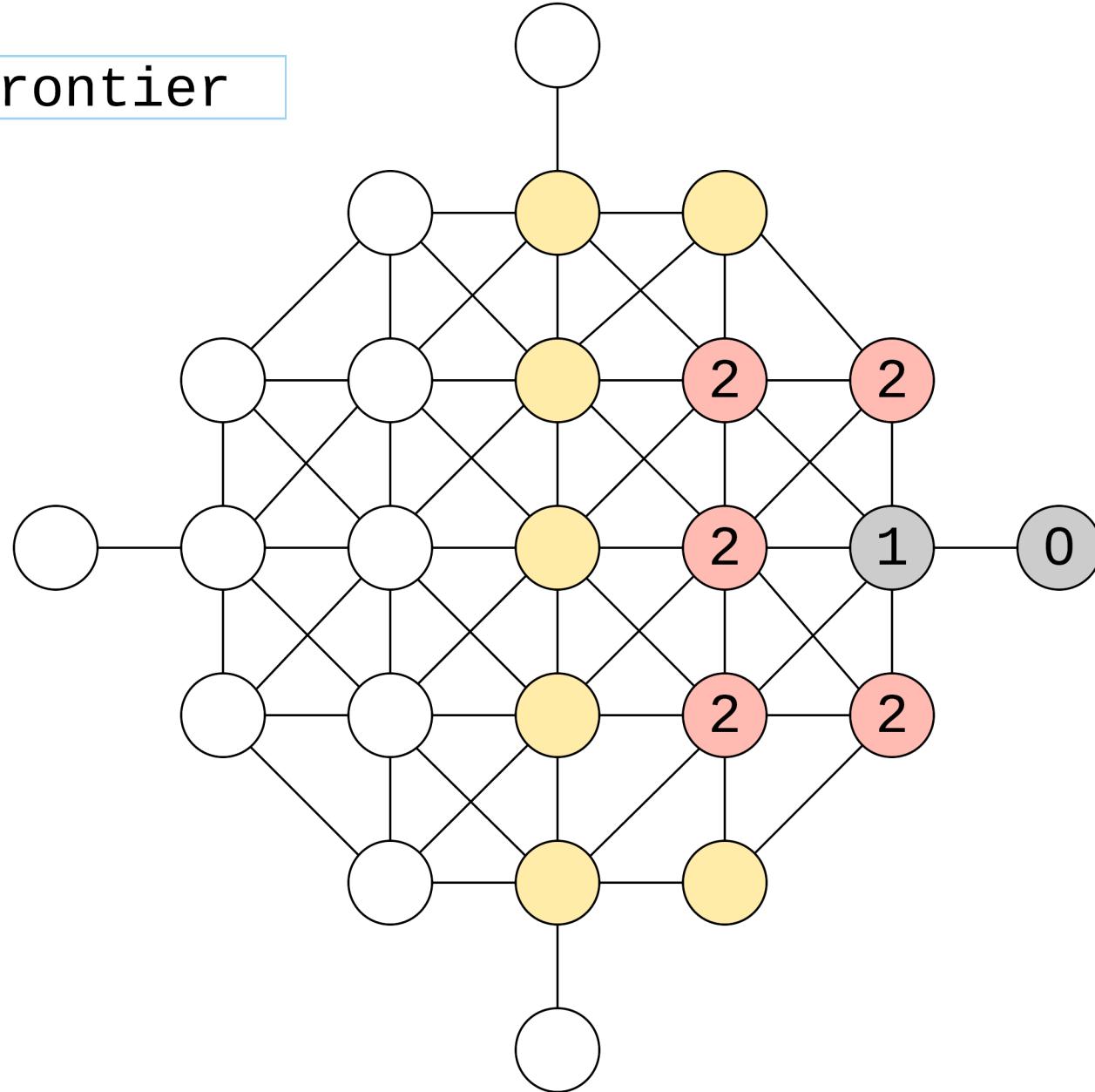


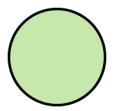


Next

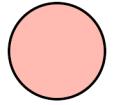


Frontier

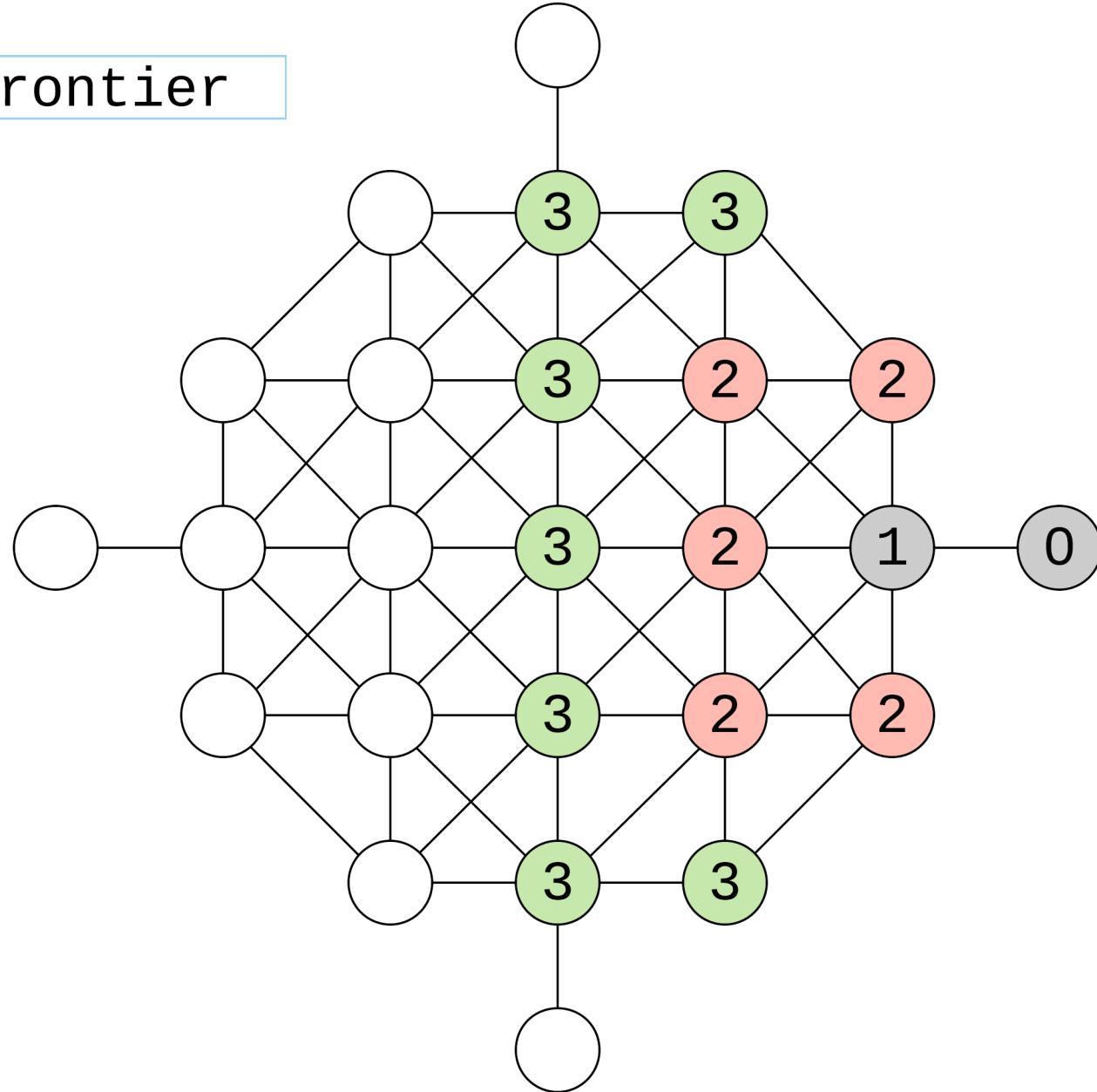


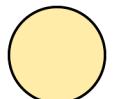


Next

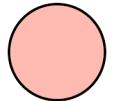


Frontier

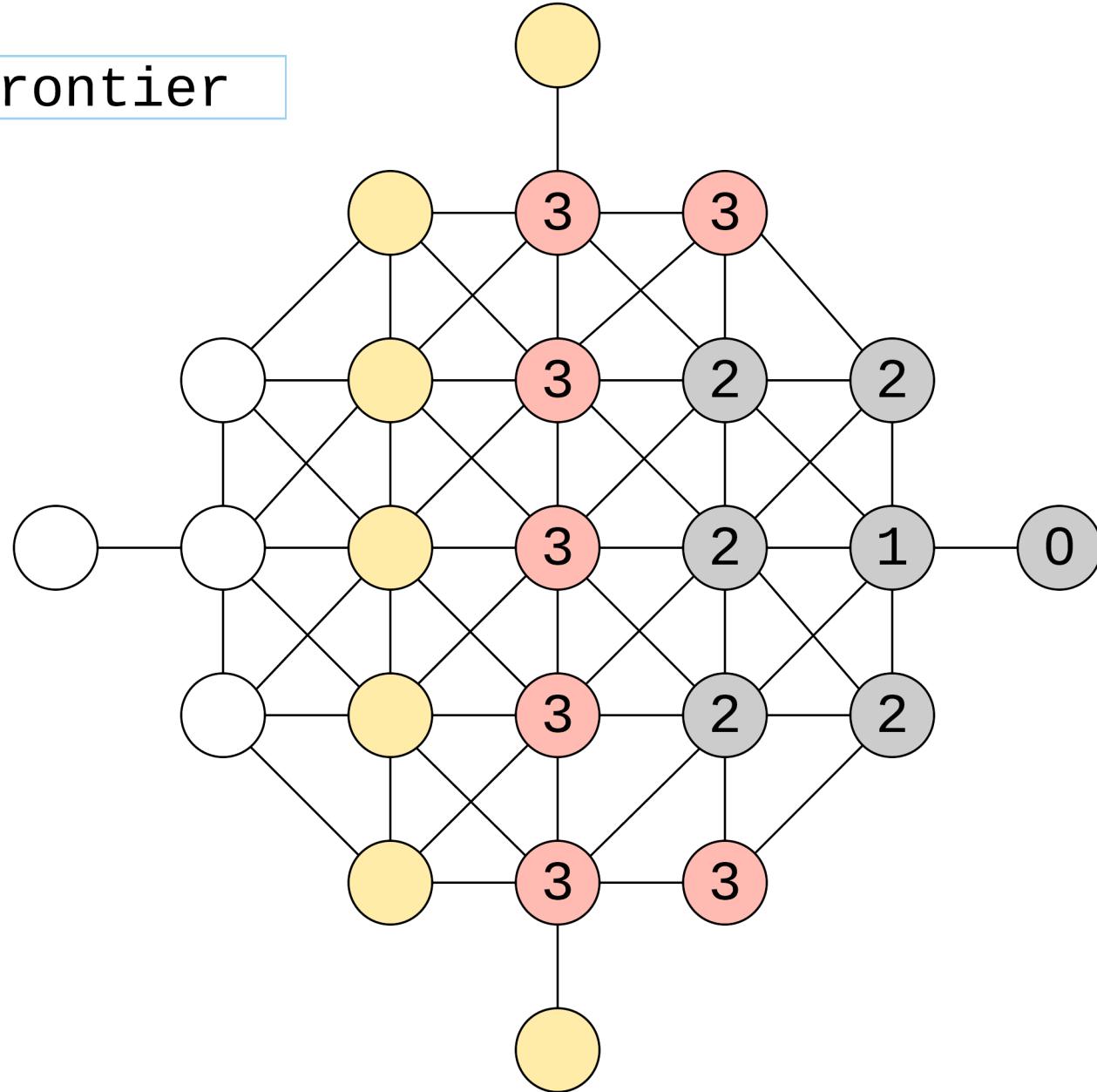


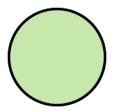


Next

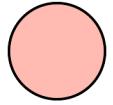


Frontier

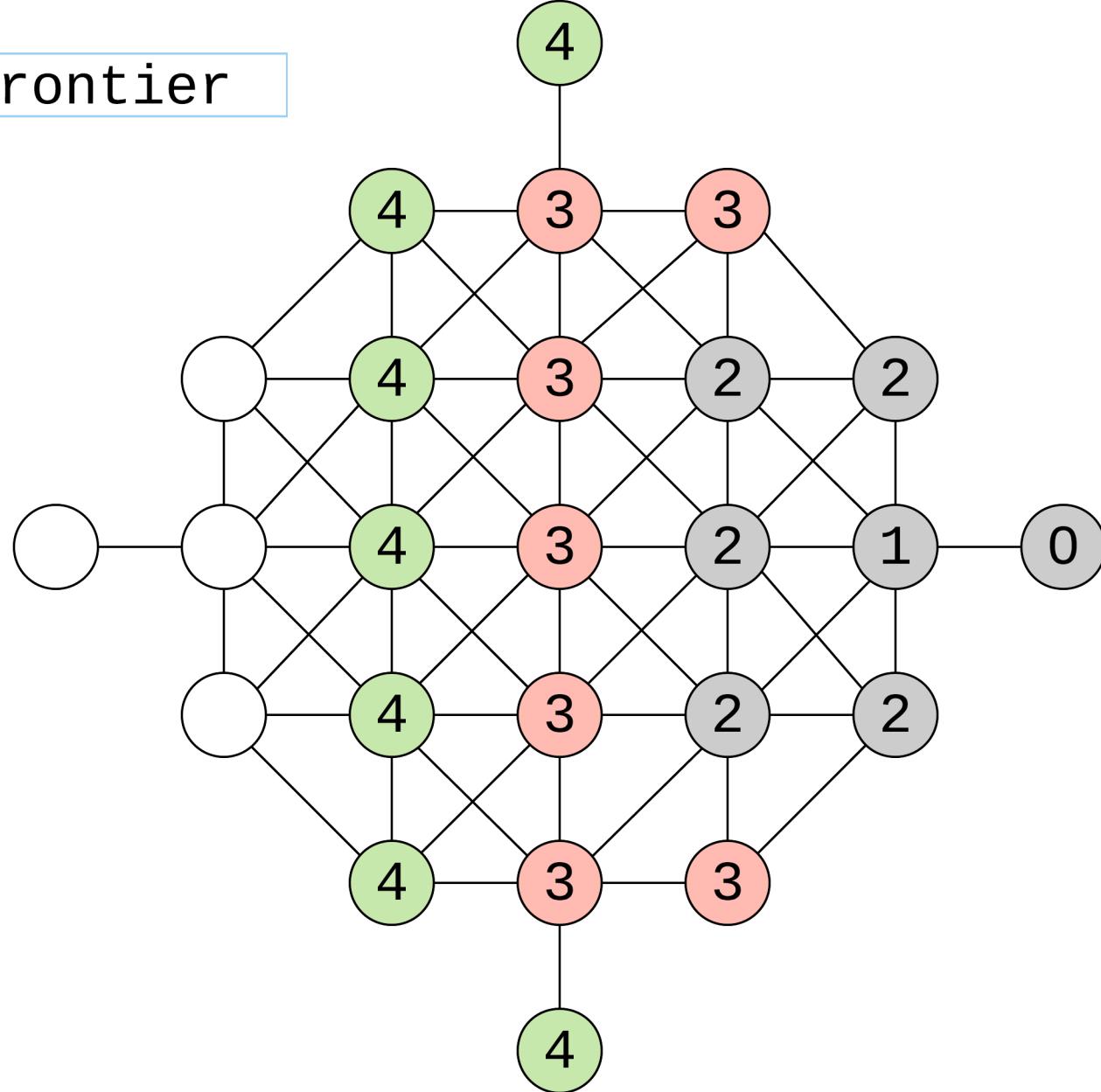


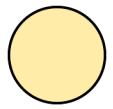


Next

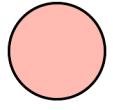


Frontier

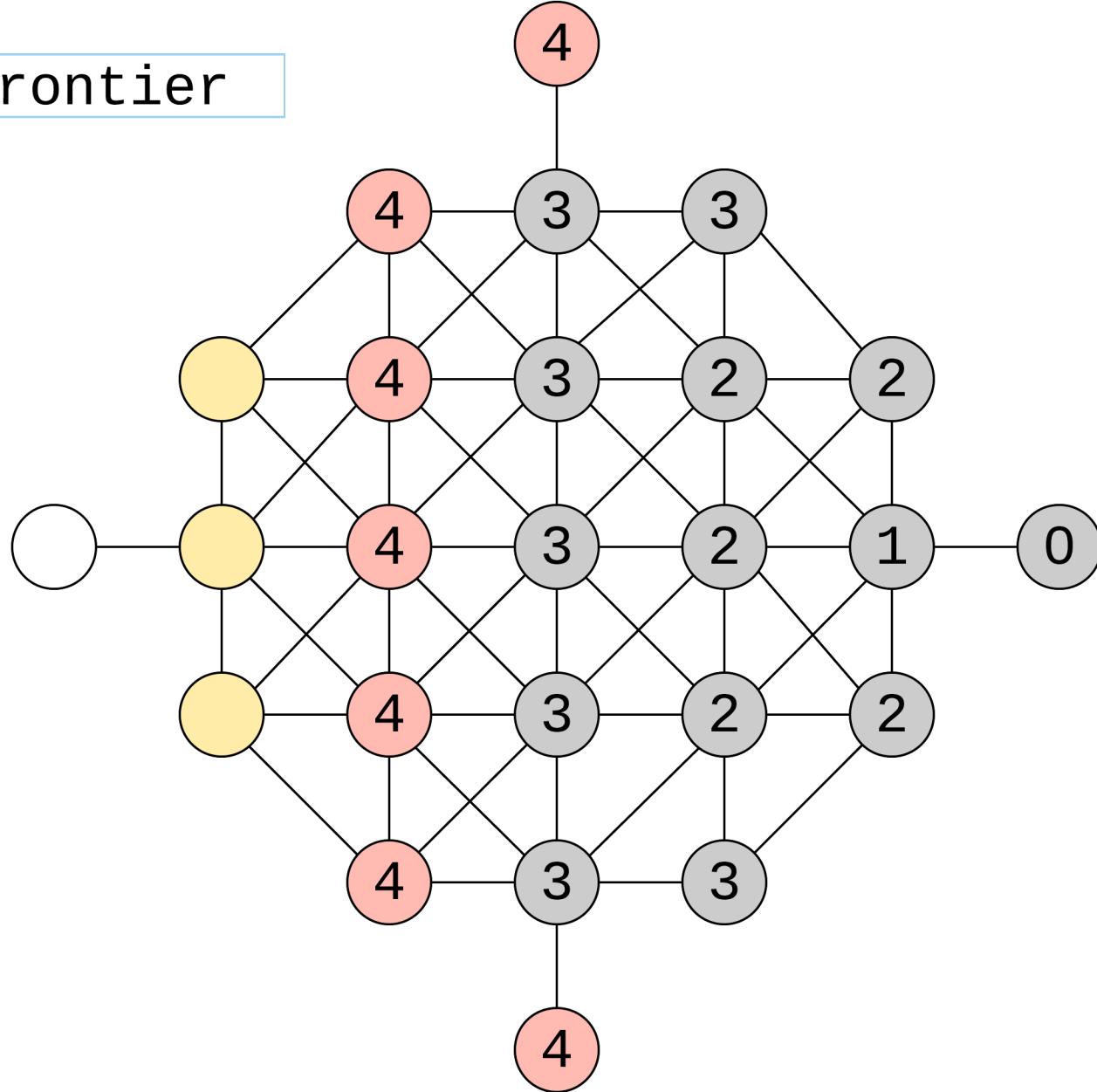


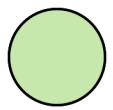


Next

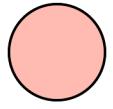


Frontier

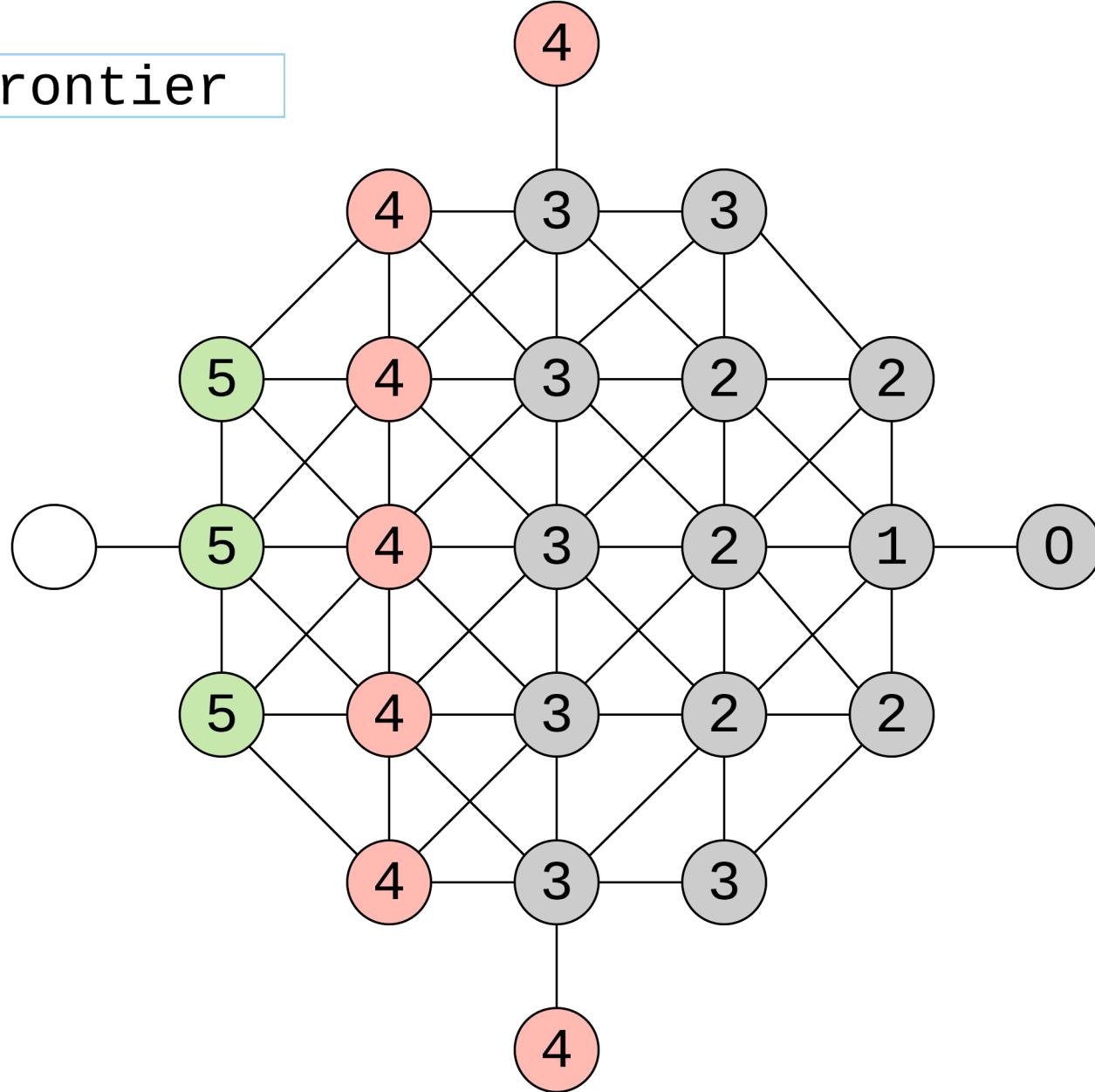


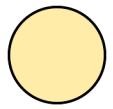


Next

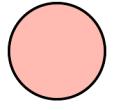


Frontier

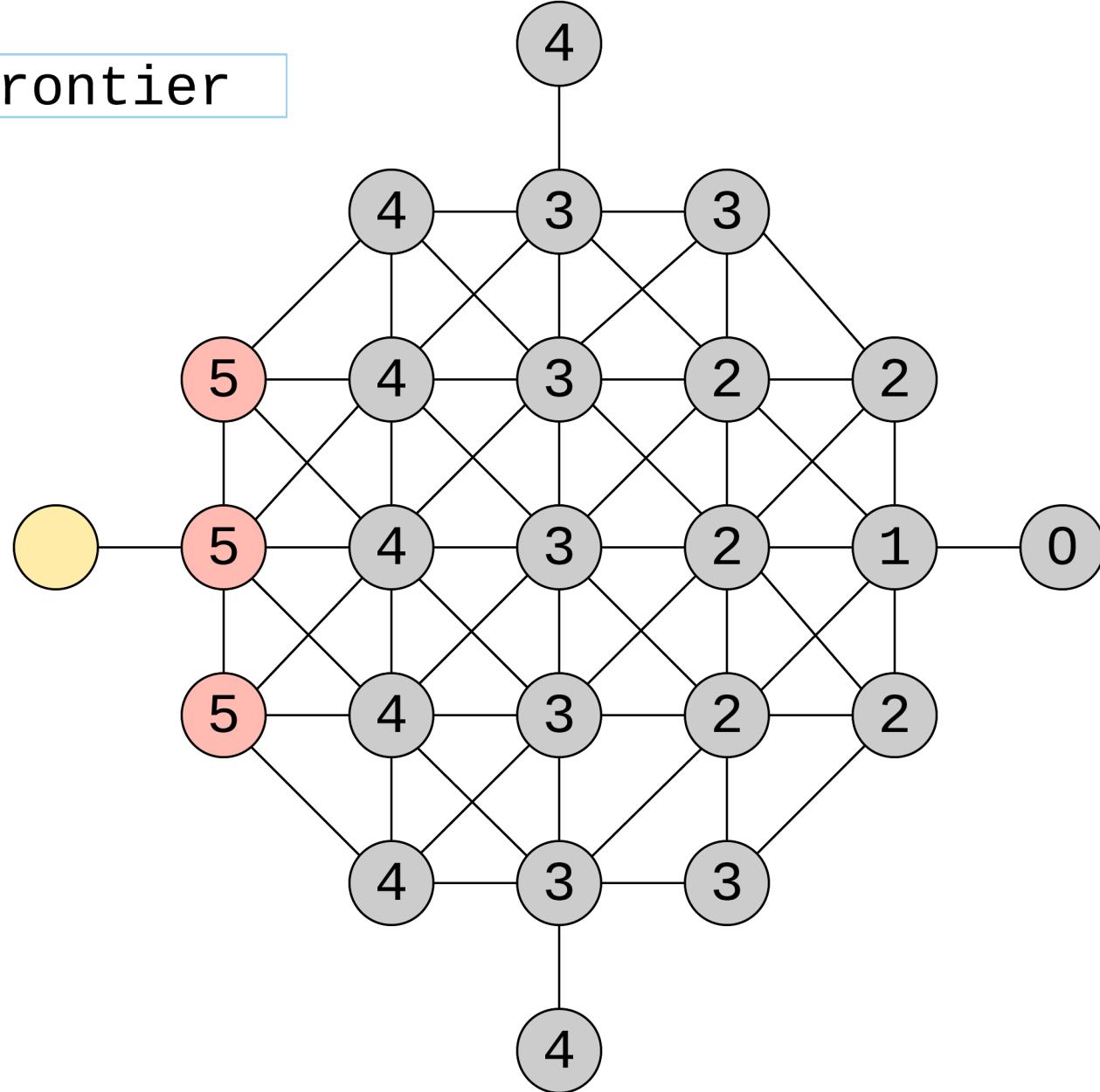


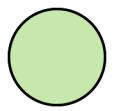


Next

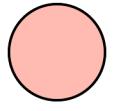


Frontier

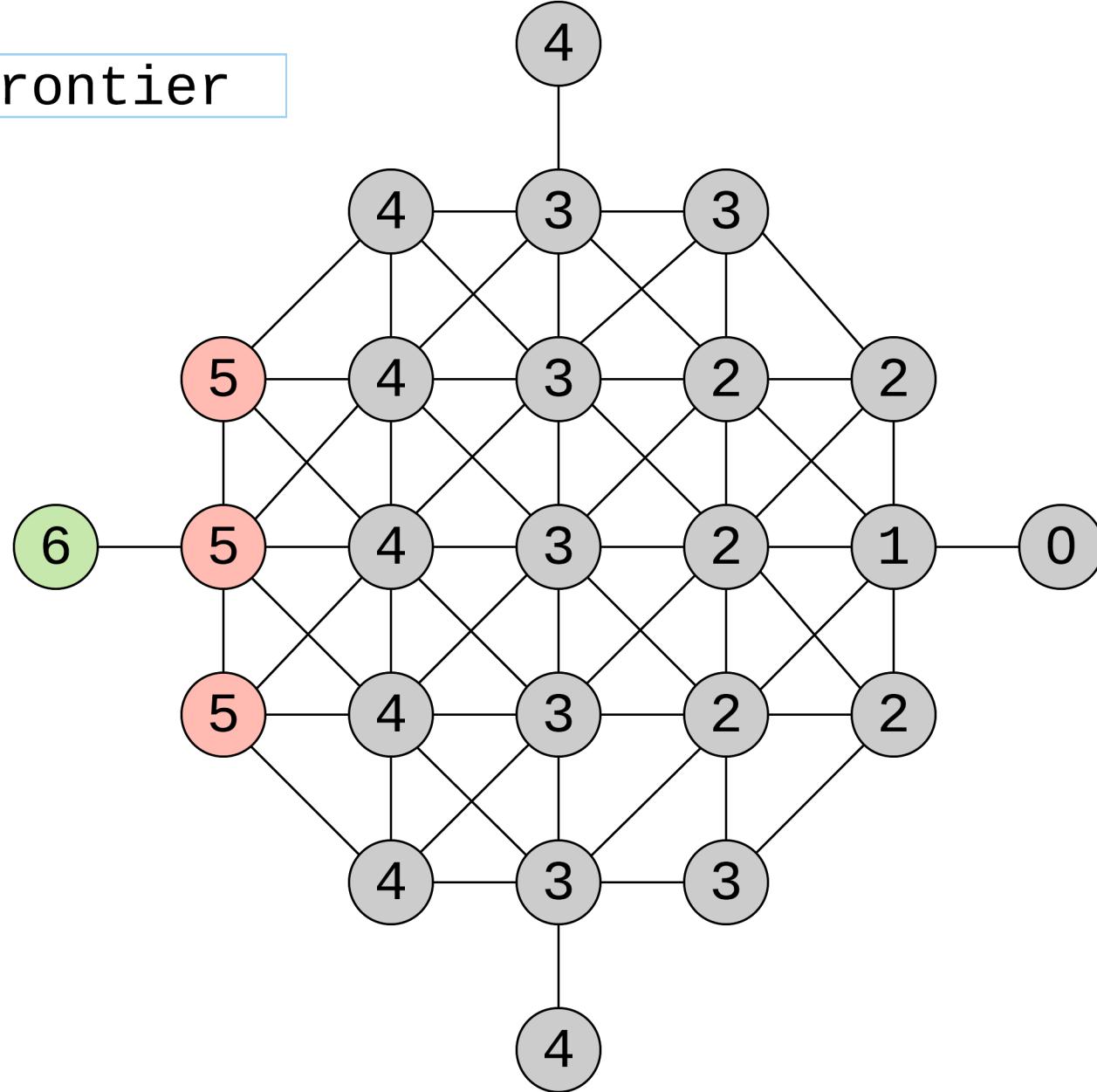


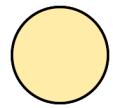


Next

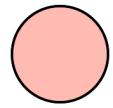


Frontier

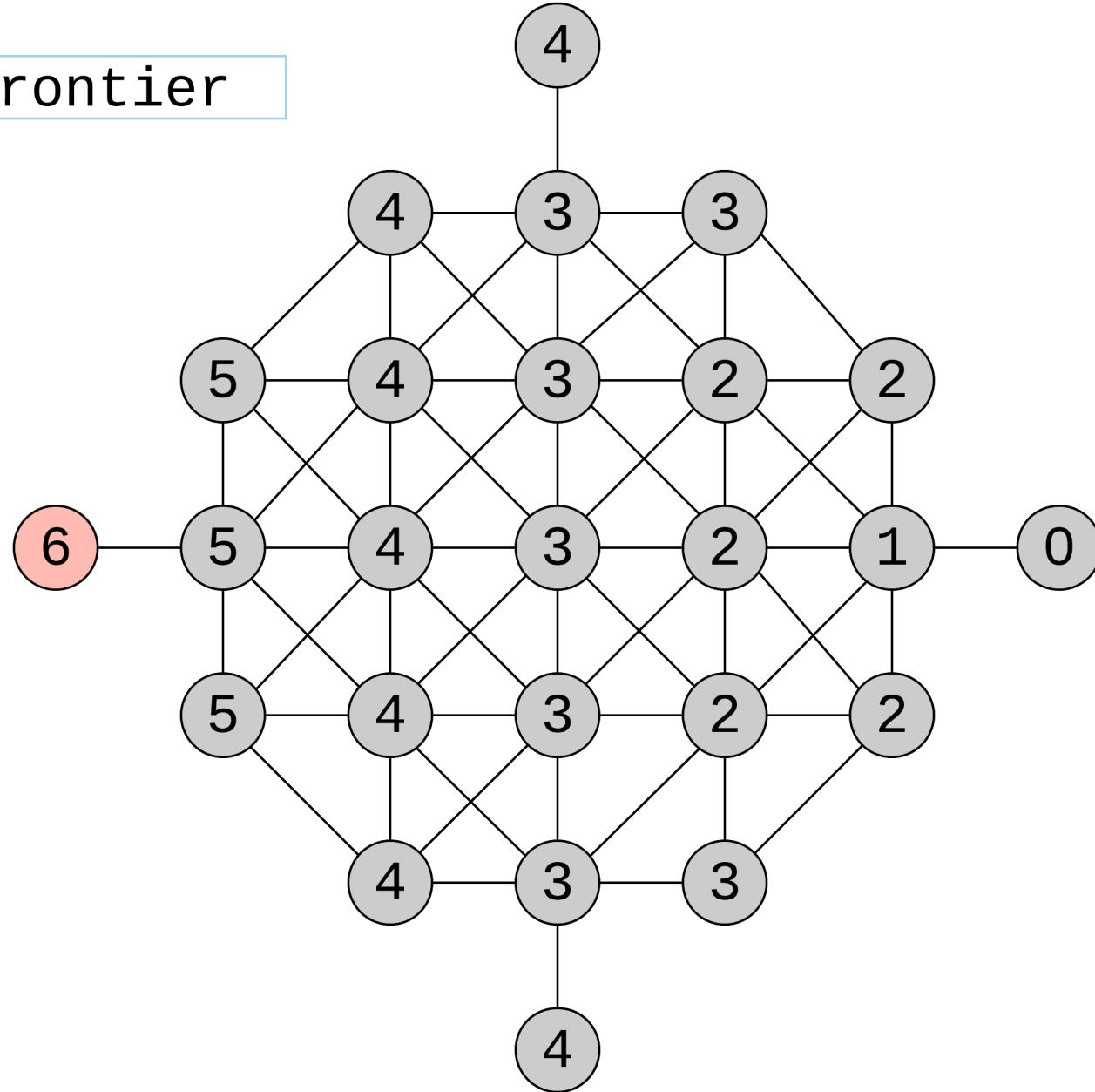


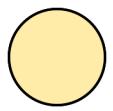


Next

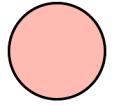


Frontier

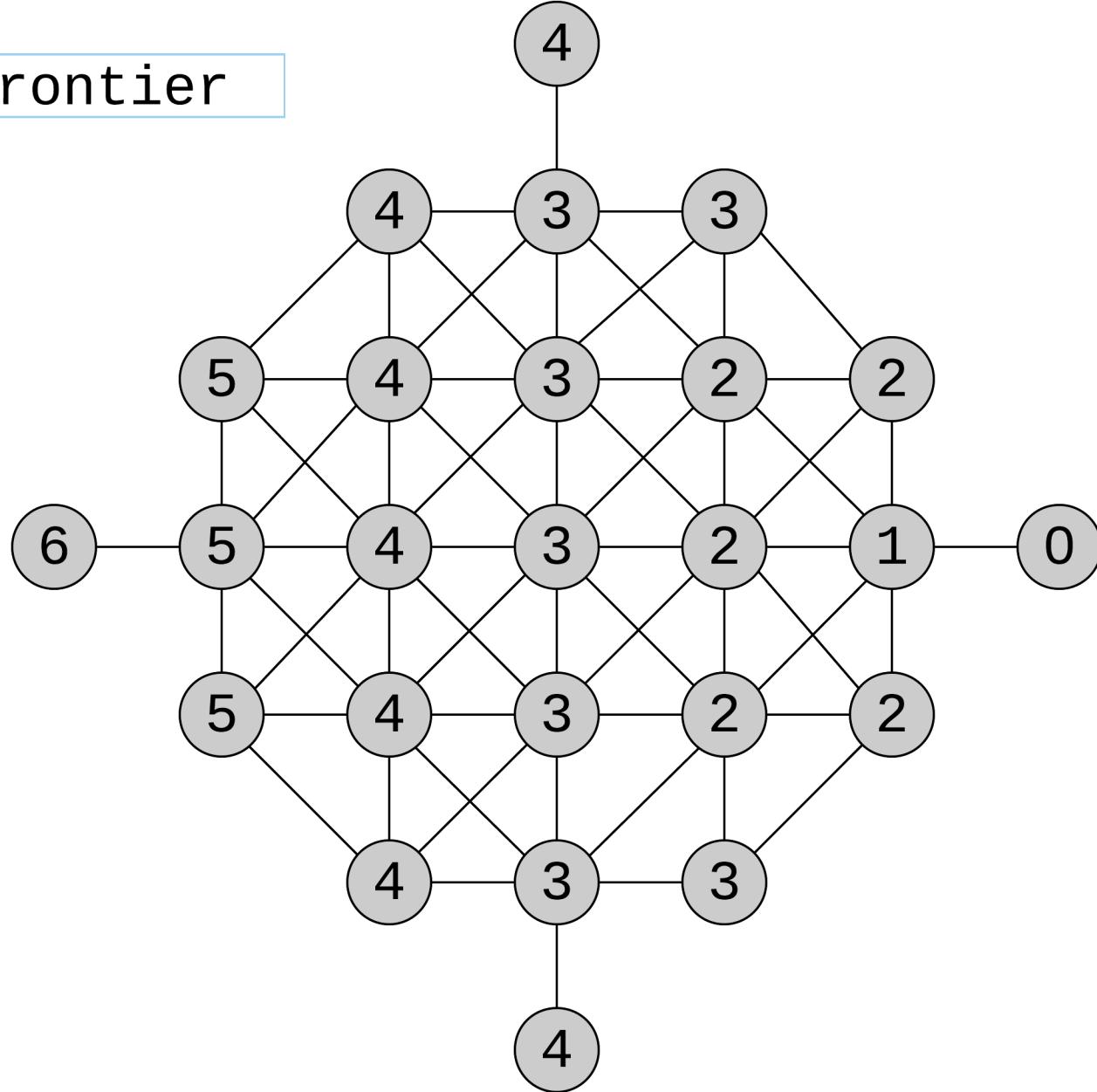


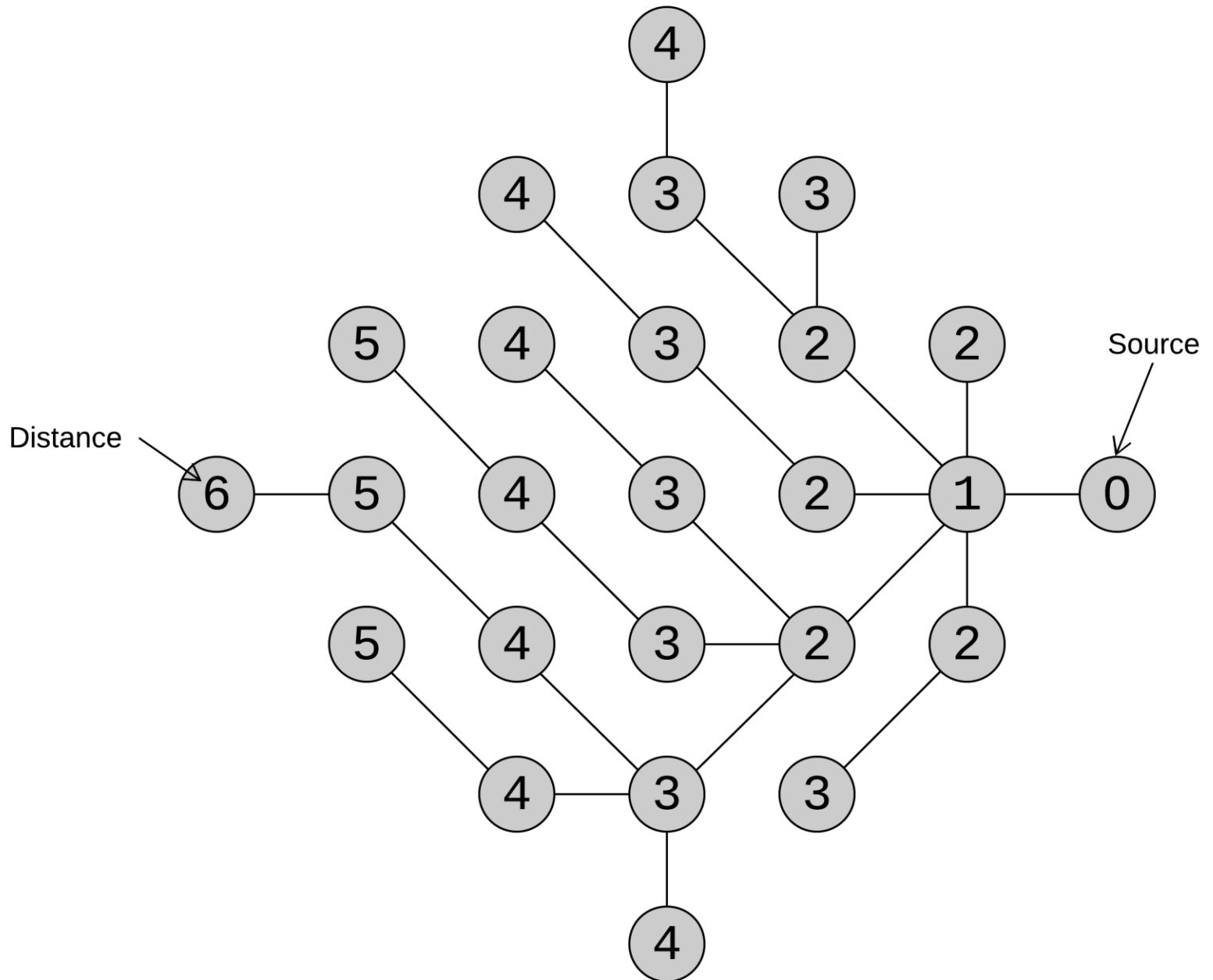


Next



Frontier

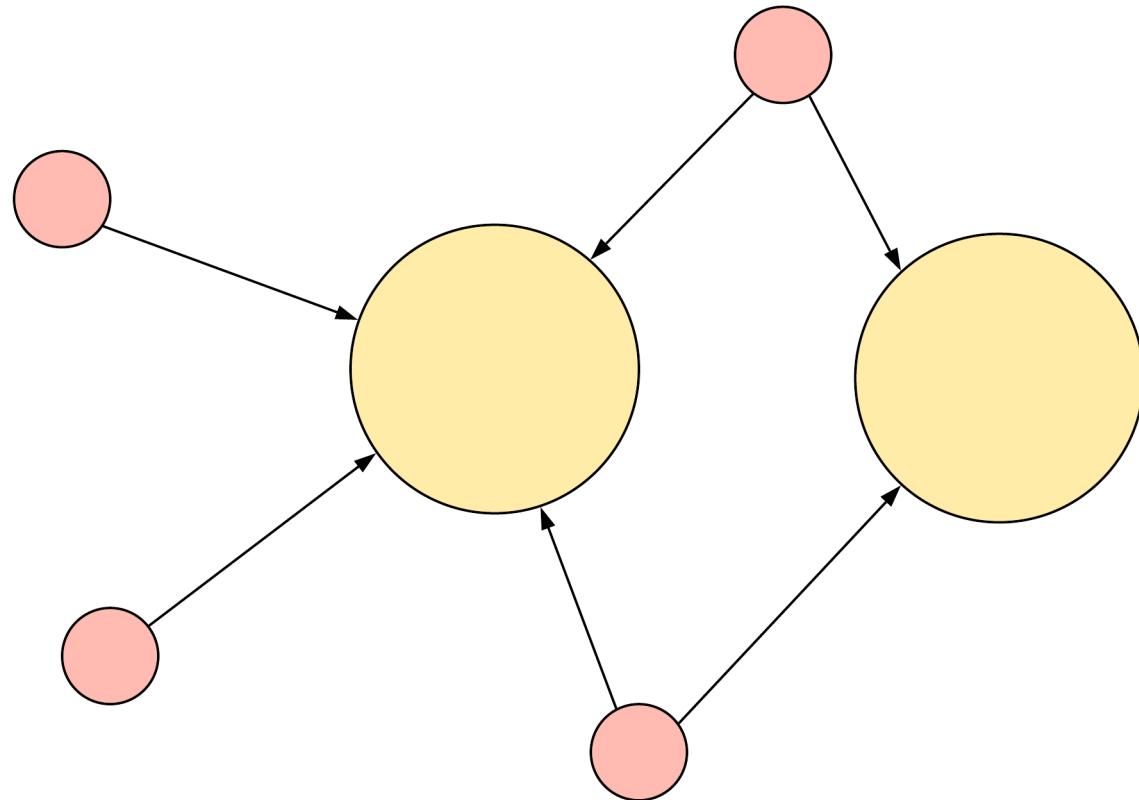




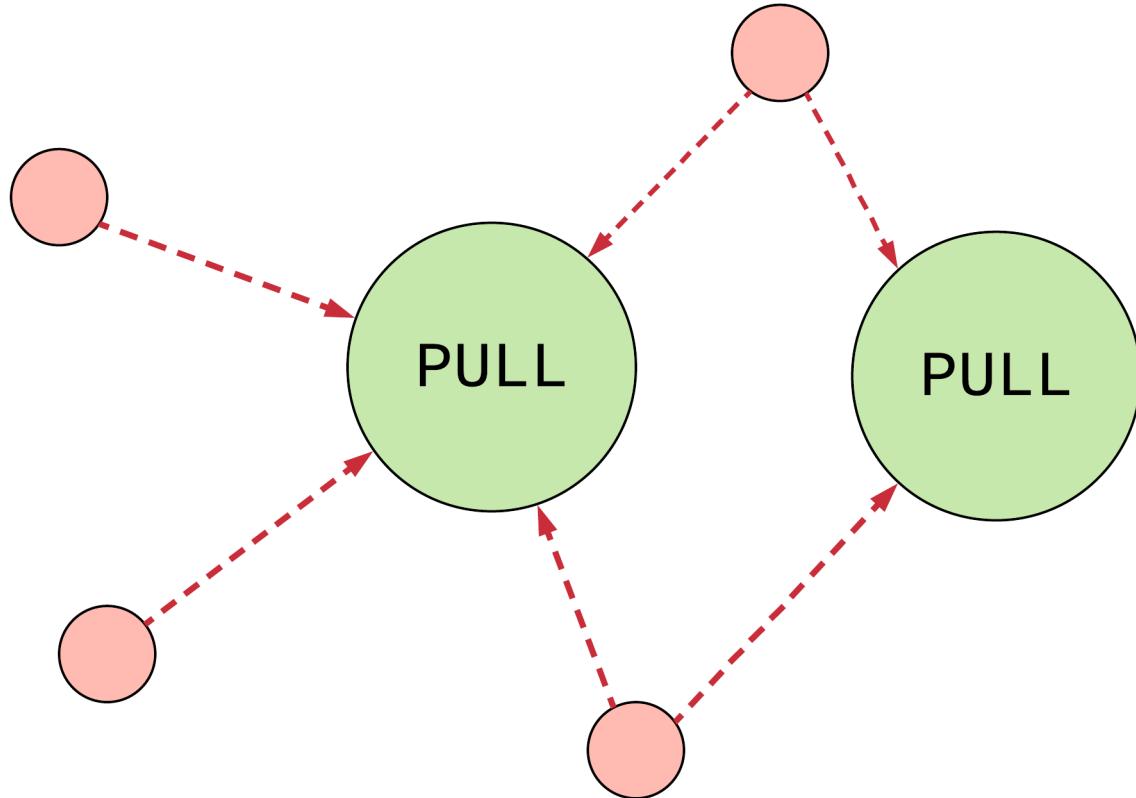
Dataflow in Graphs

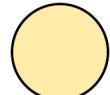
- Push: nodes are pushing data *to* other nodes.
- Pull: nodes are pulling data *from* other nodes.

Pull

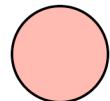


Pull

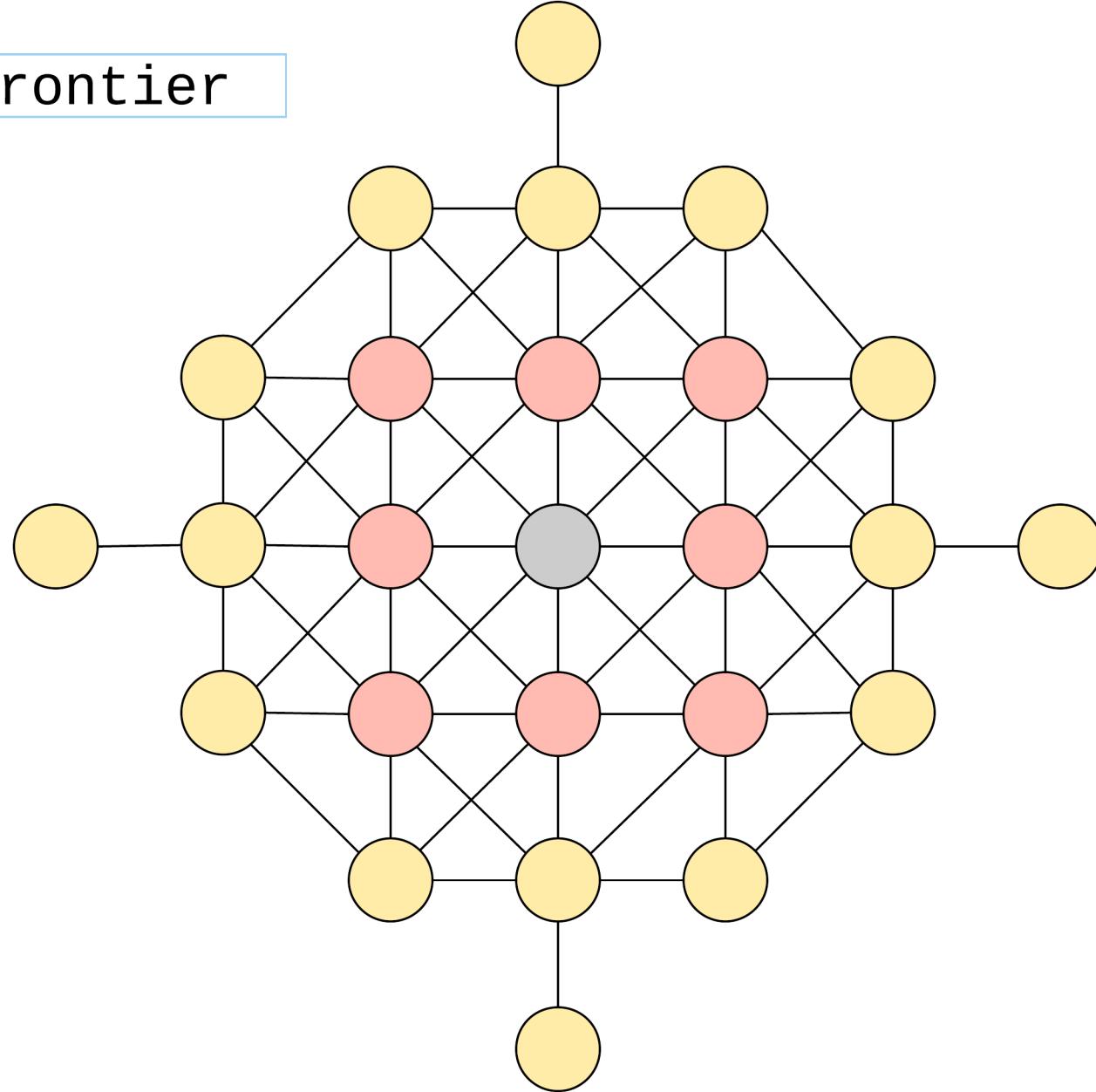


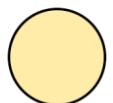


Next

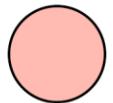


Frontier

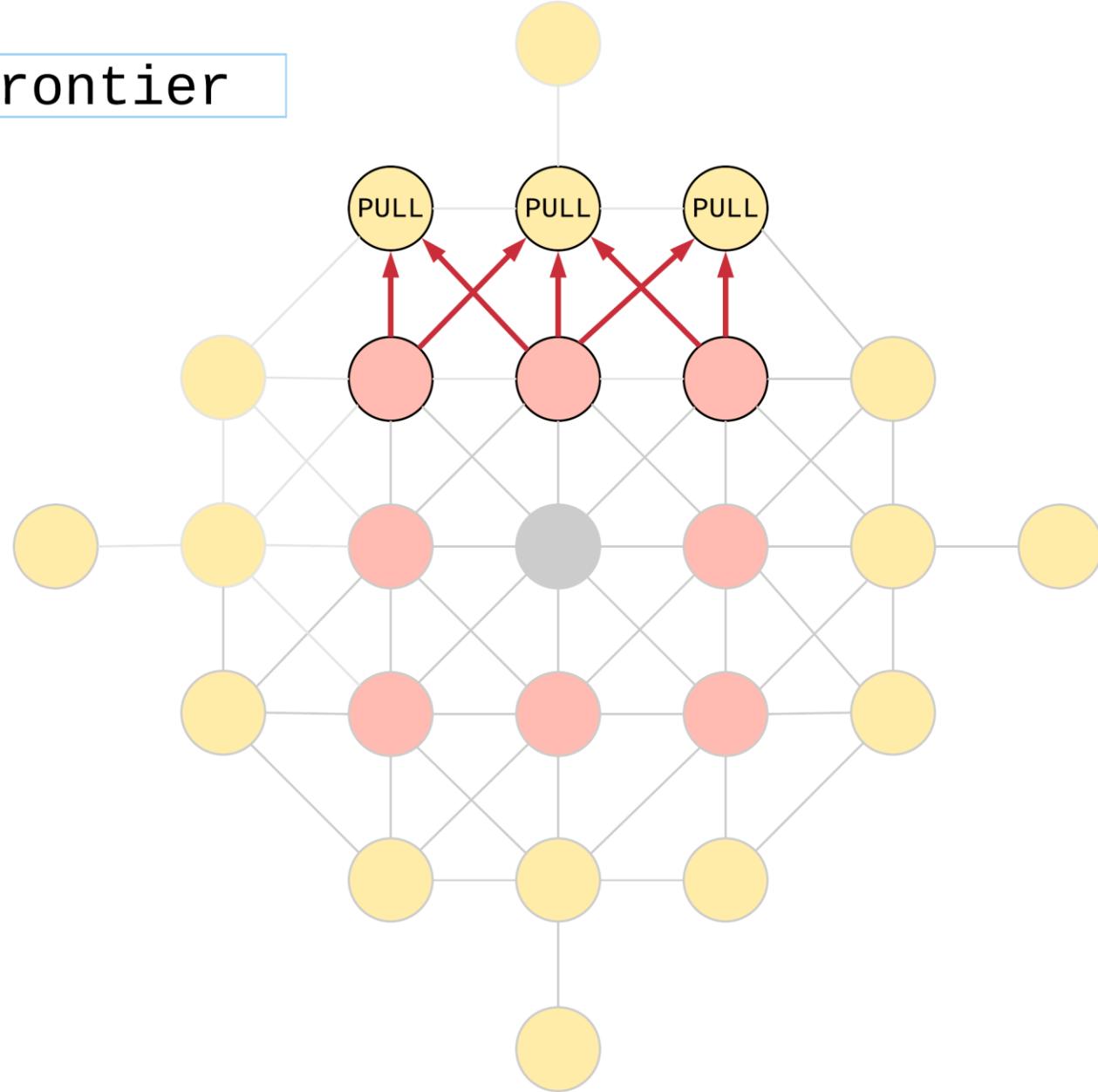


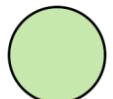


Next

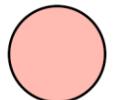


Frontier

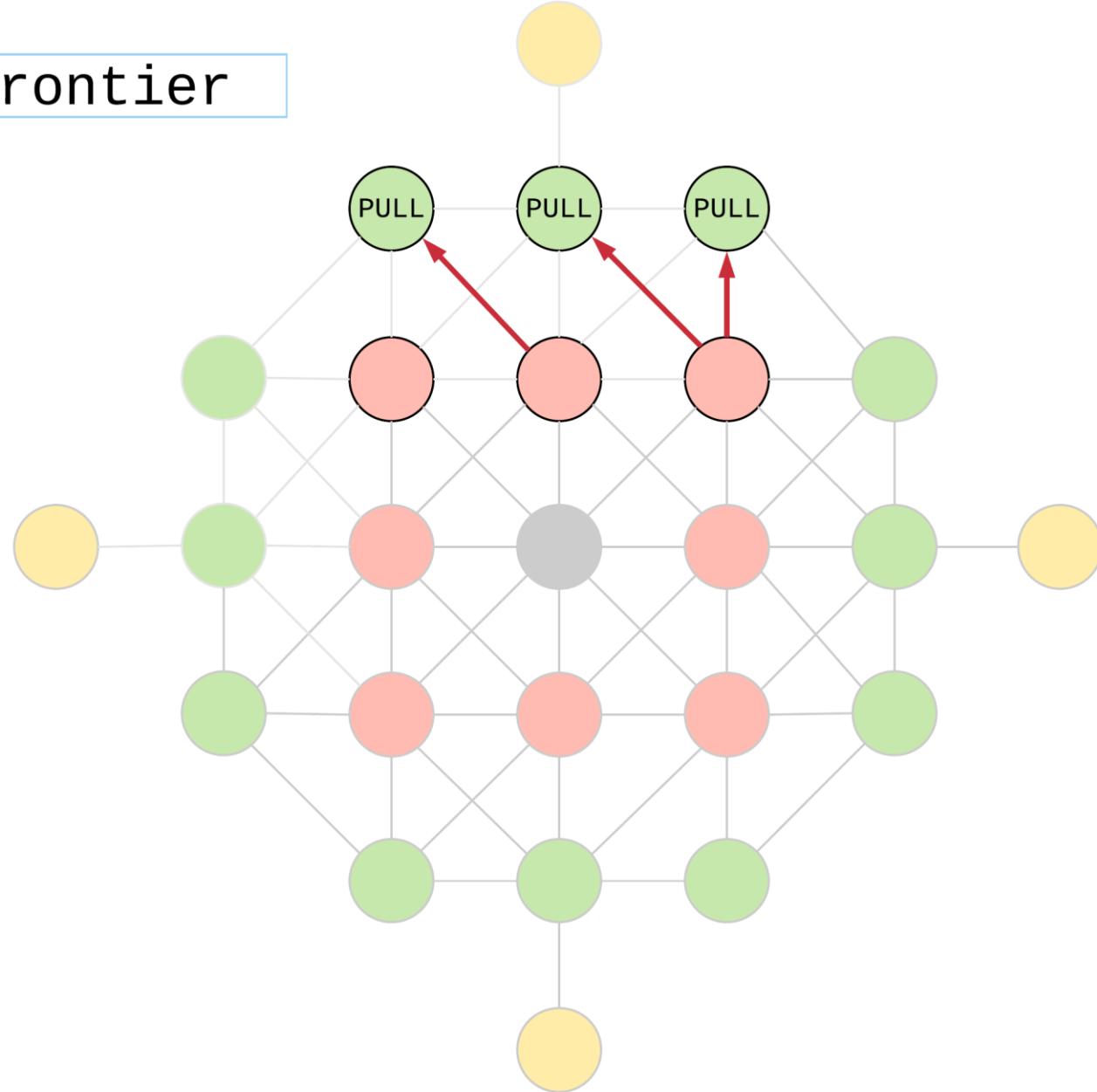




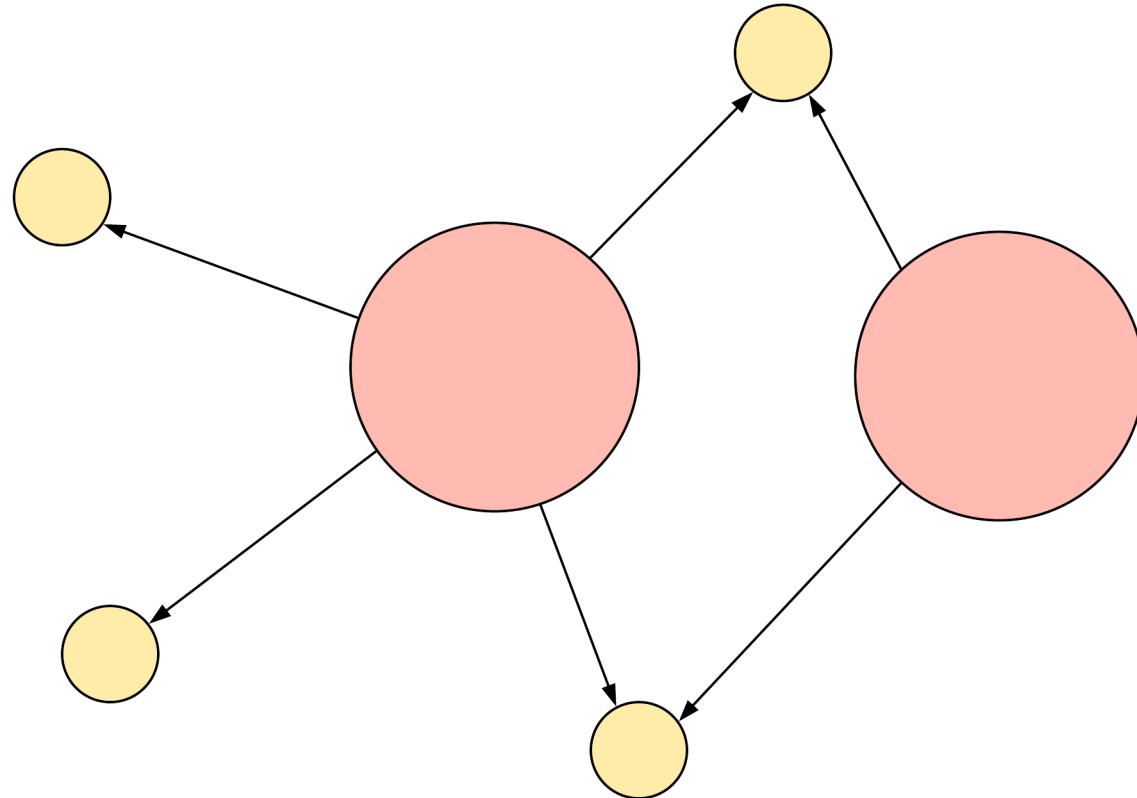
Next



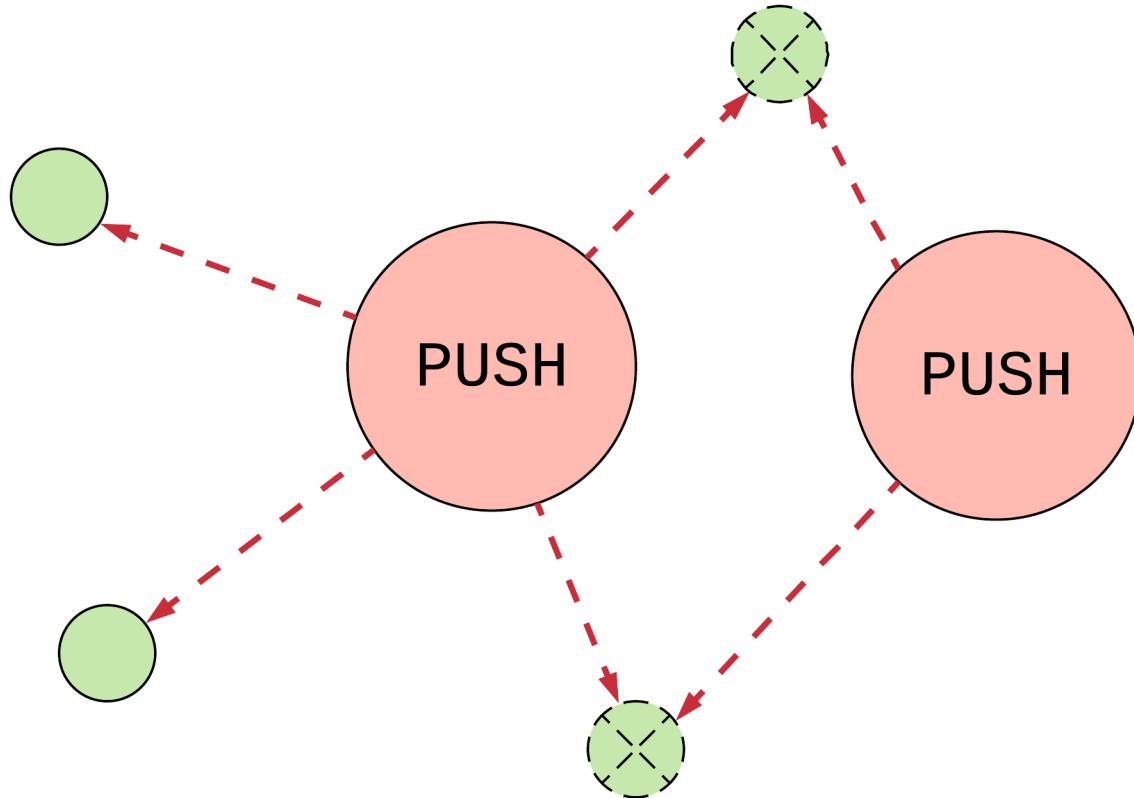
Frontier



Push

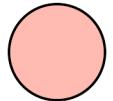


Push

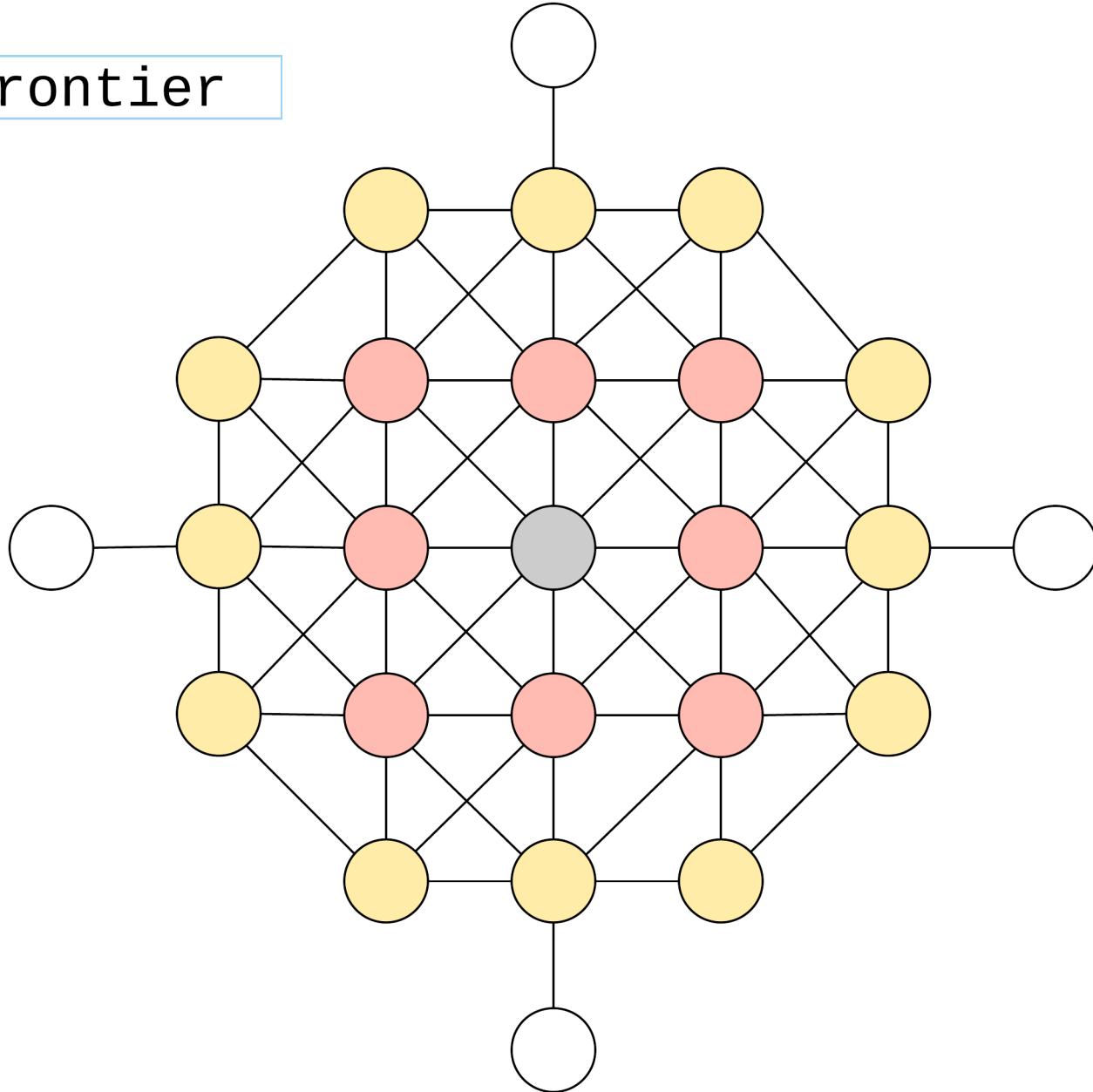




Next

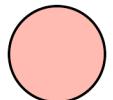


Frontier

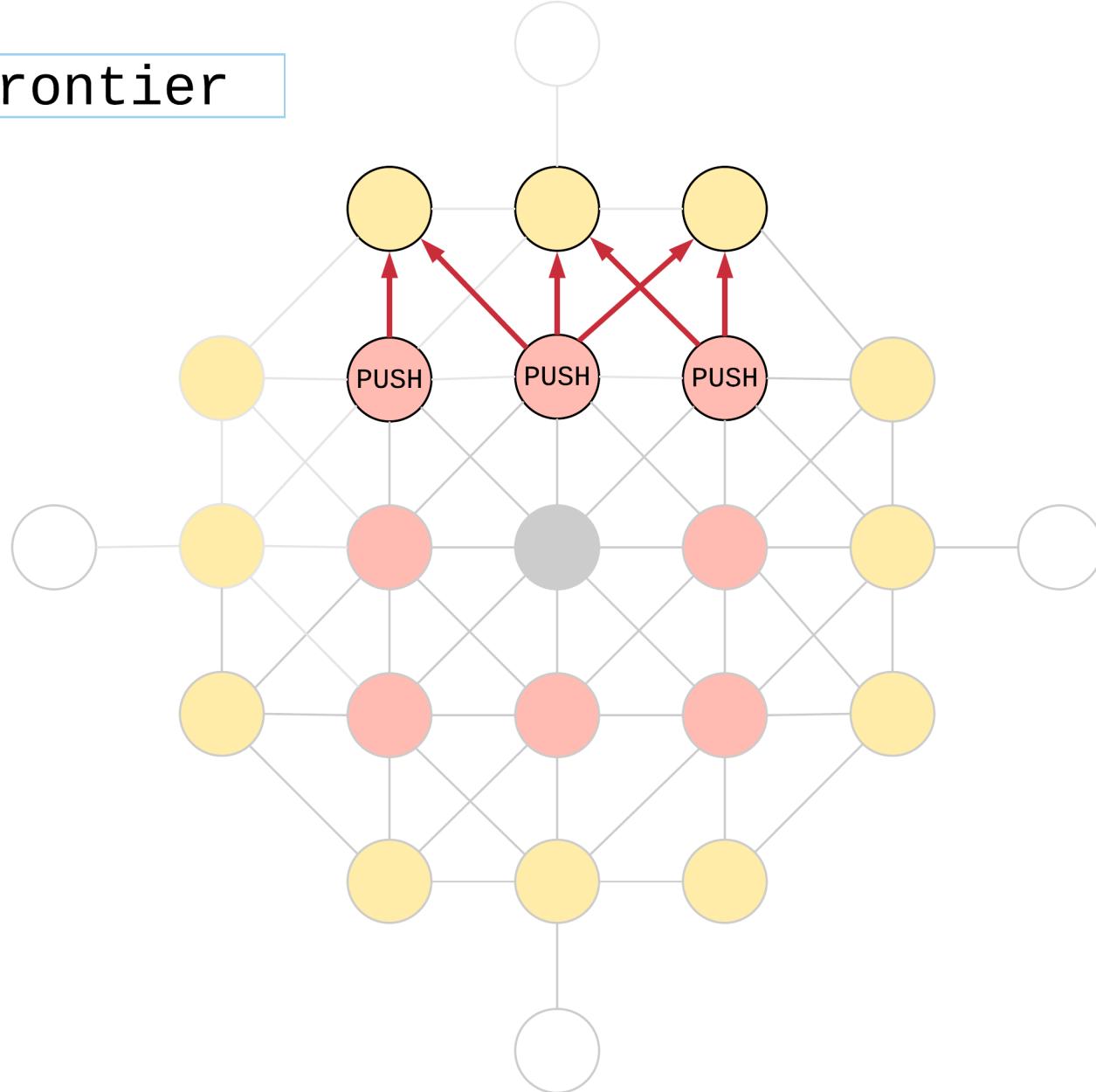


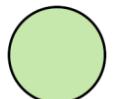


Next

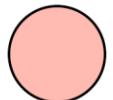


Frontier

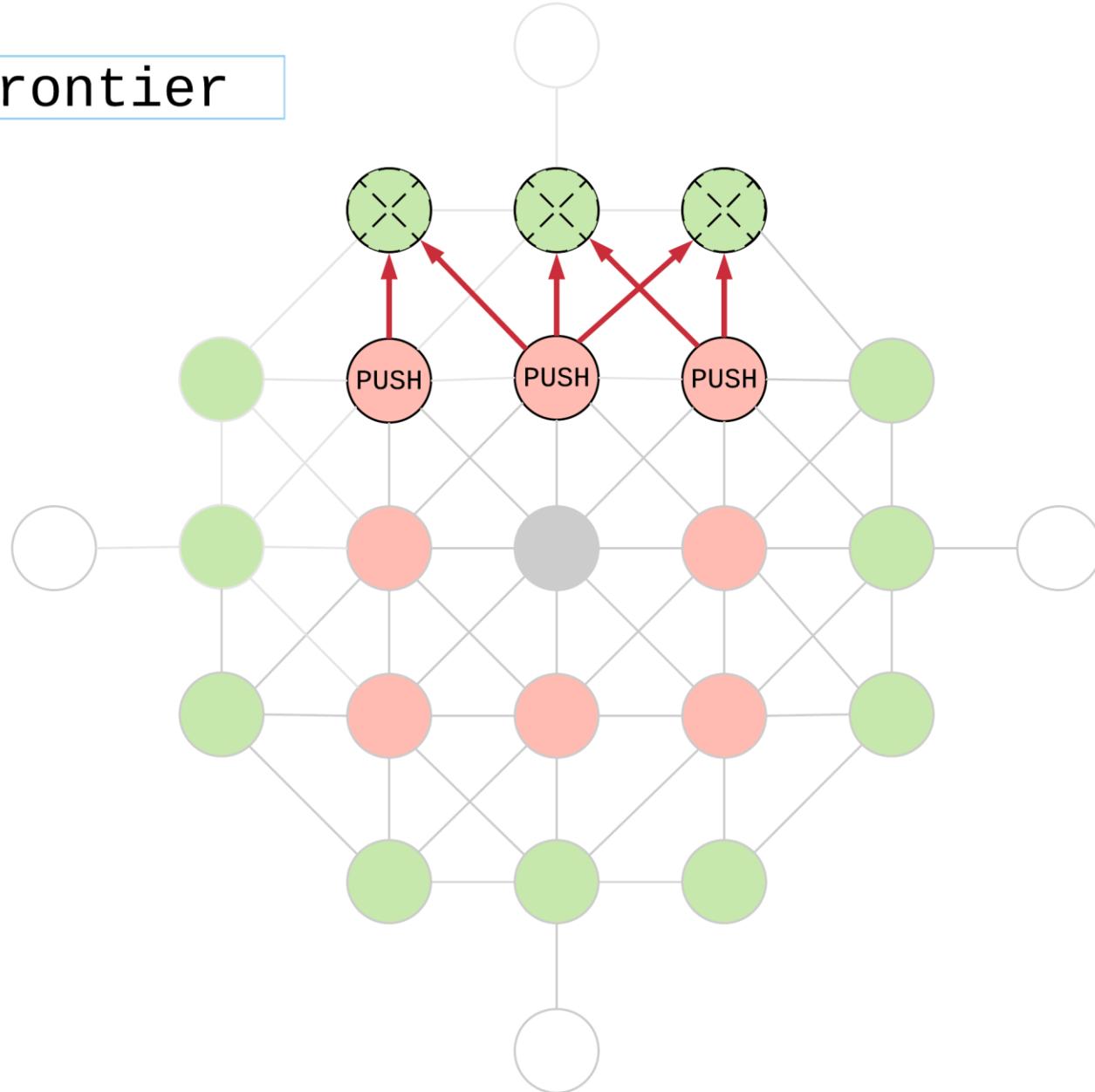




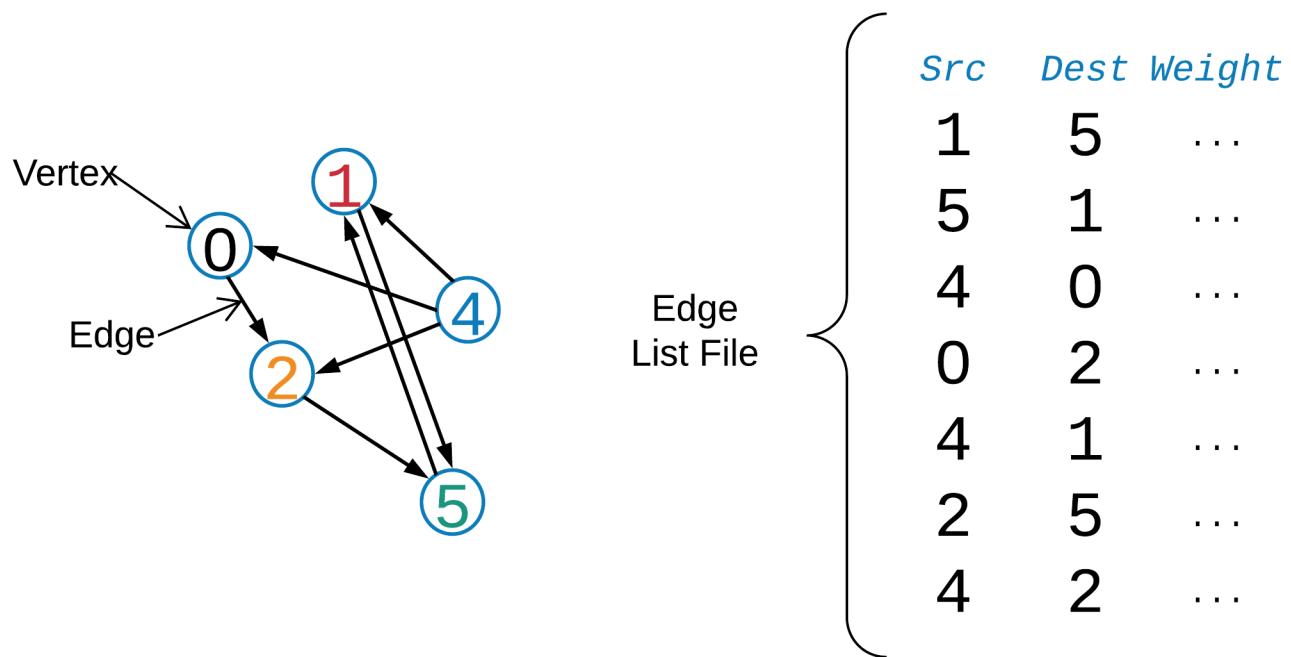
Next



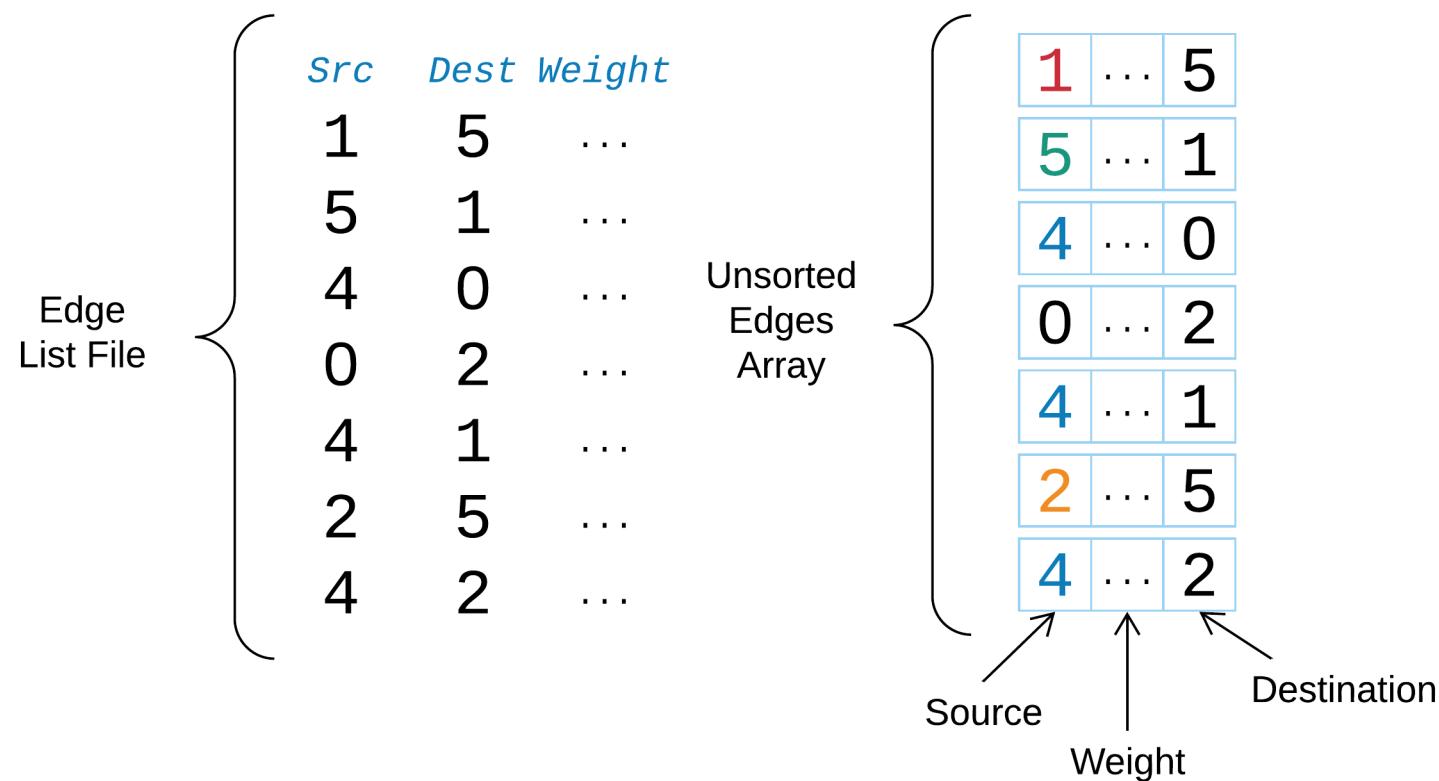
Frontier



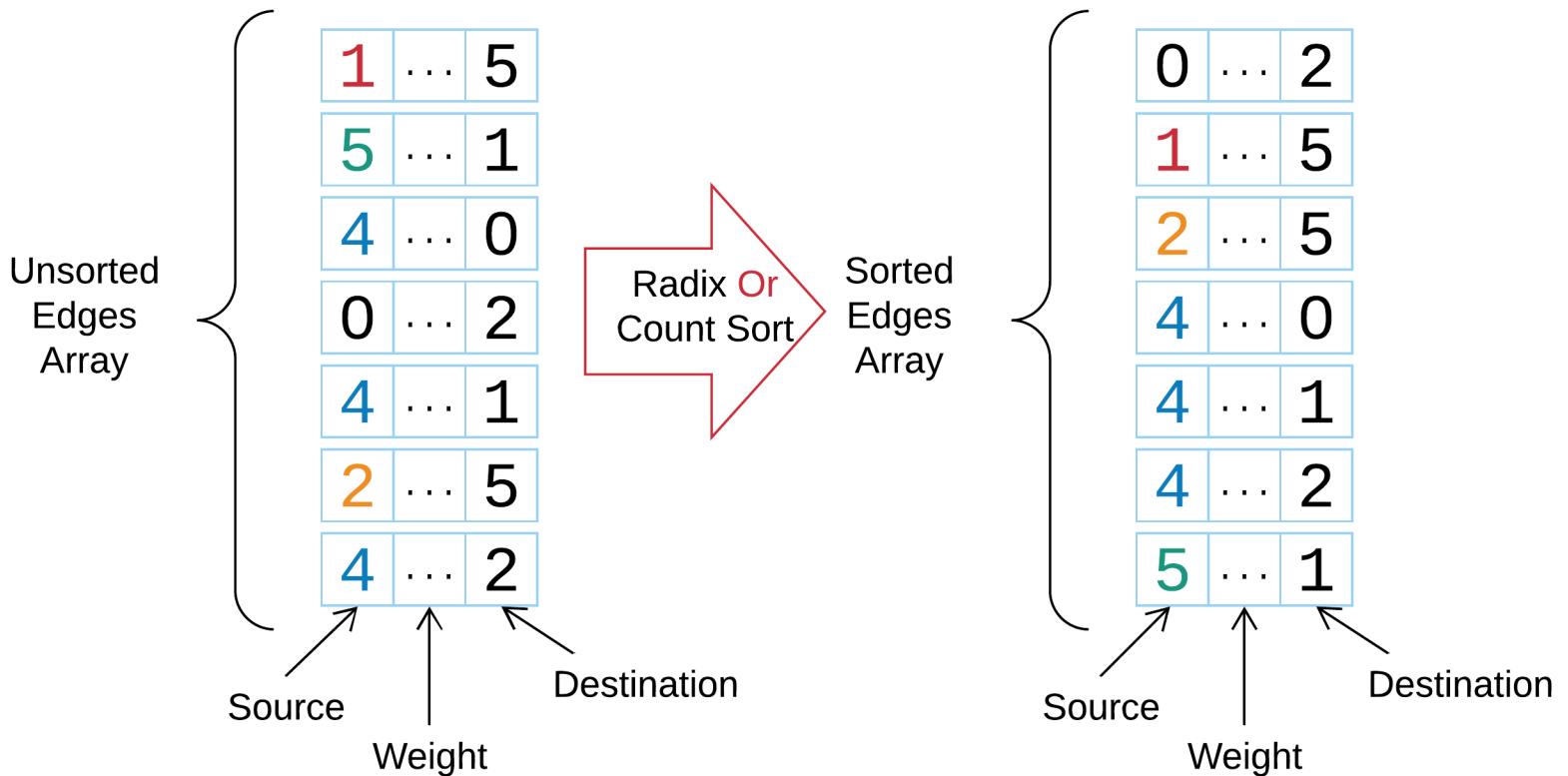
Recap



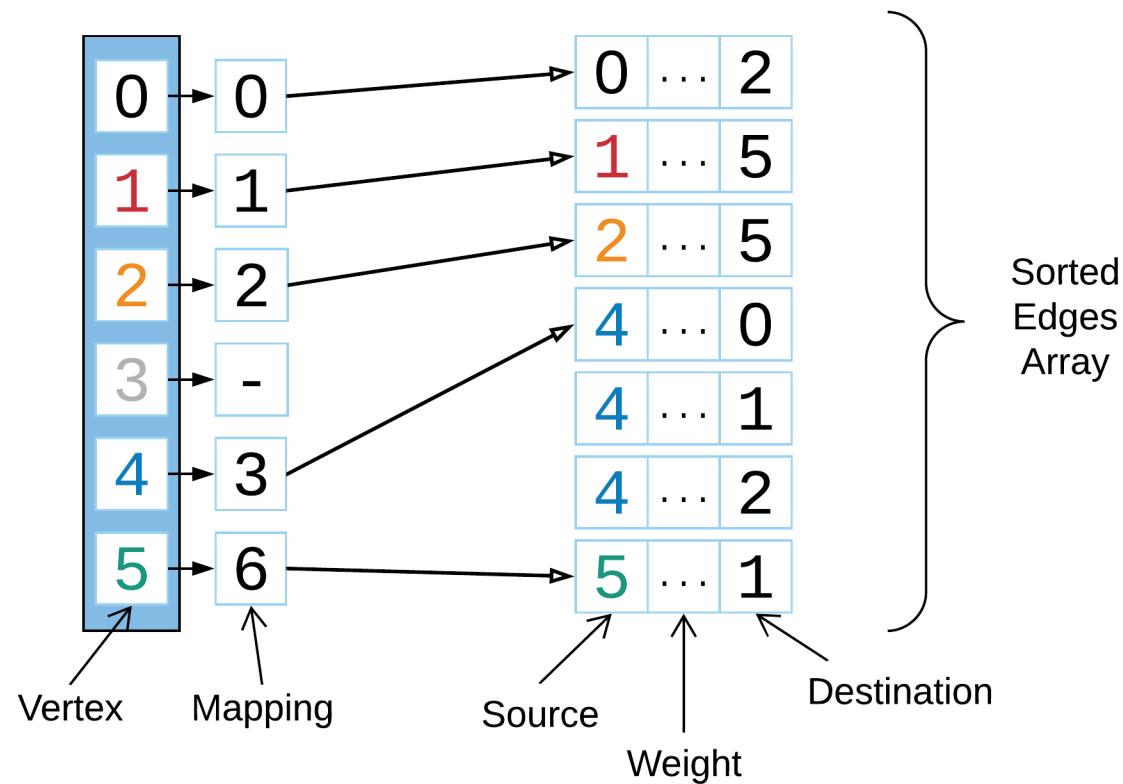
Load edge list file



Sort edge array



Map to vertices

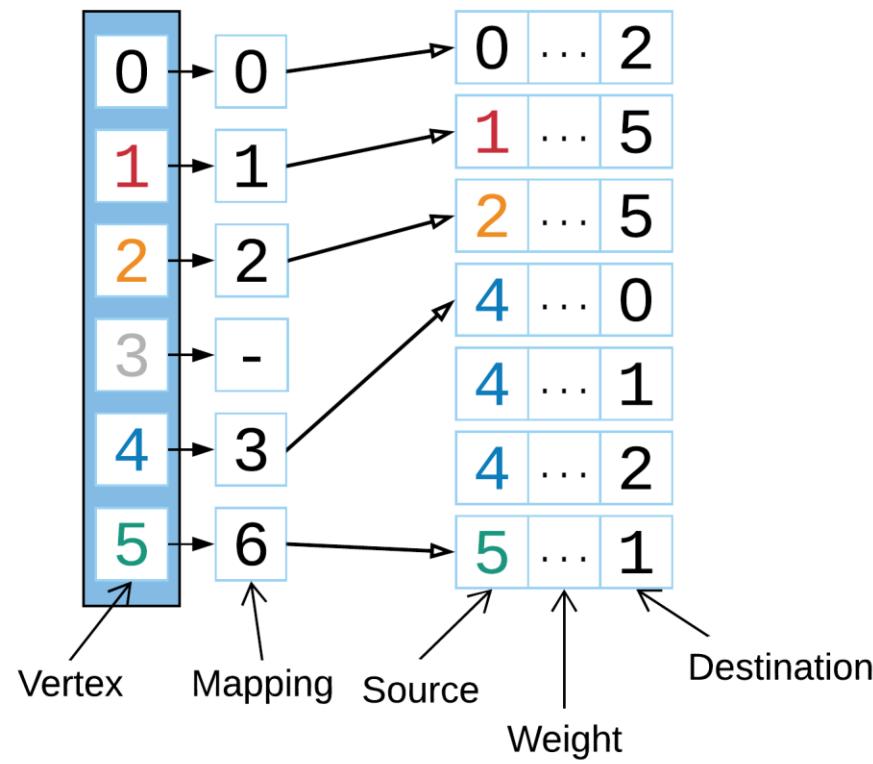
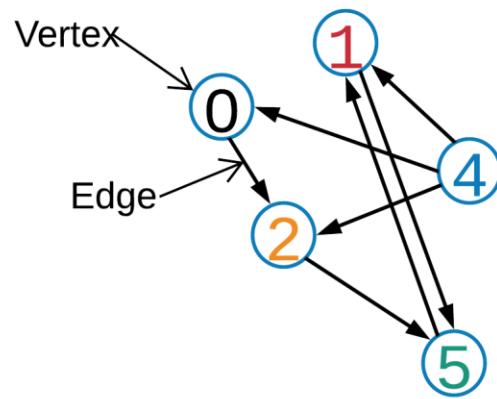


BFS Top-Down-Approach (Push)

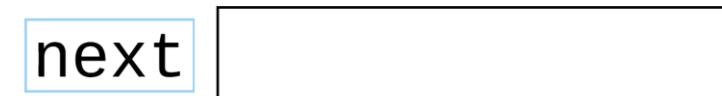
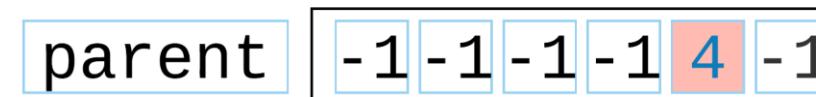
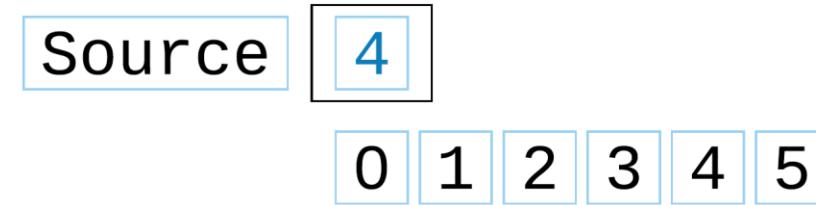
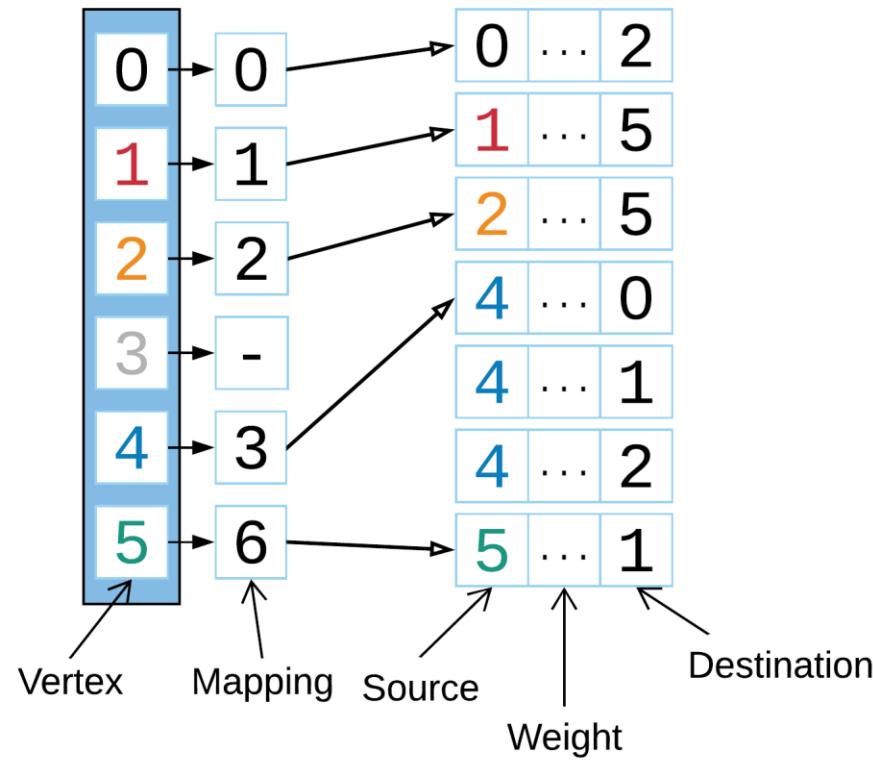
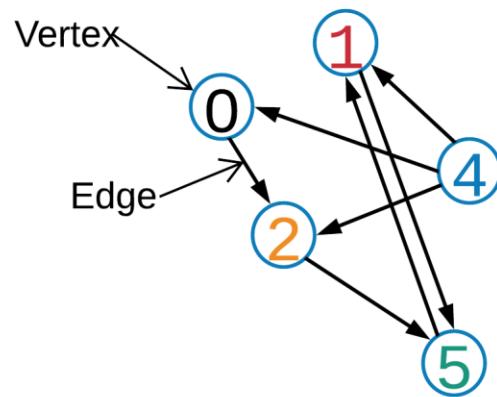
```
function breadth-first-search(vertices, source)
    frontier ← {source}
    next ← {}
    parents ← [-1, -1, ..., -1]
    while frontier ≠ {} do
        top-down-step(vertices, frontier, next, parents)
        frontier ← next
        next ← {}
    end while
    return tree
```

BFS Top-Down-Approach (Push)

```
function top-down-step(vertices, frontier, next, parents)
    for v ∈ frontier do
        for n ∈ neighbors[v] do
            if parents[n] = -1 then
                parents[n] ← v
                next ← next ∪ {n}
            end if
        end for
    end for
```

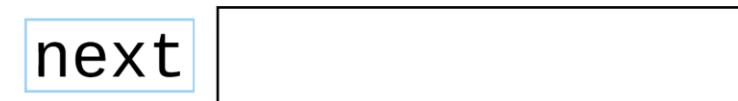
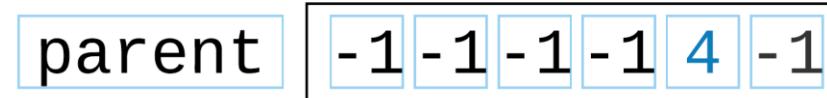
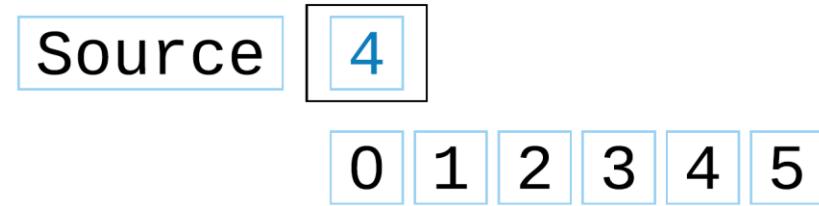
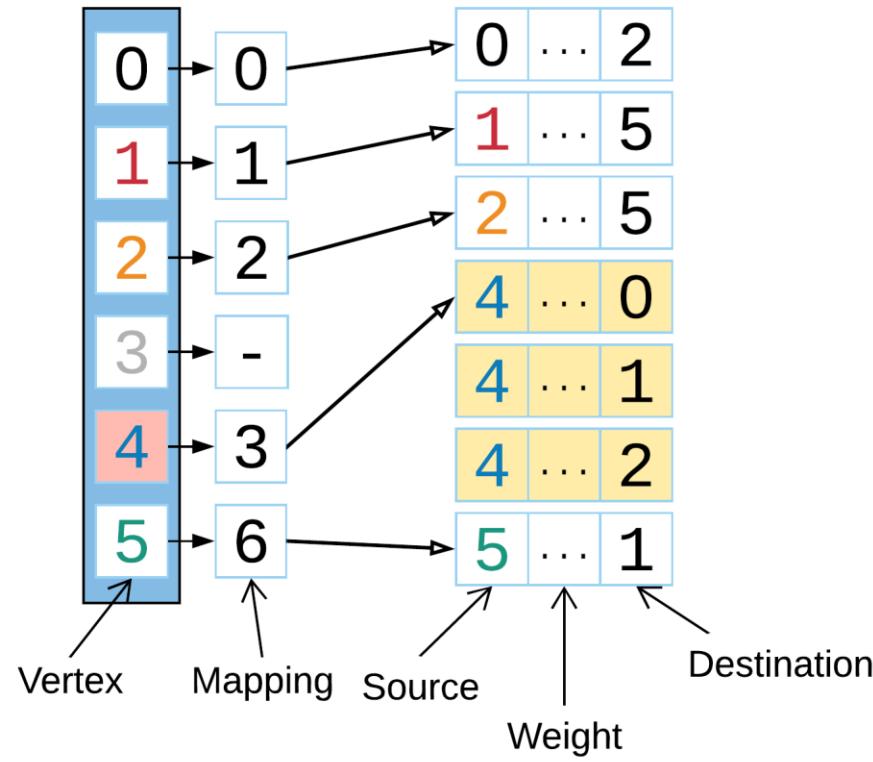
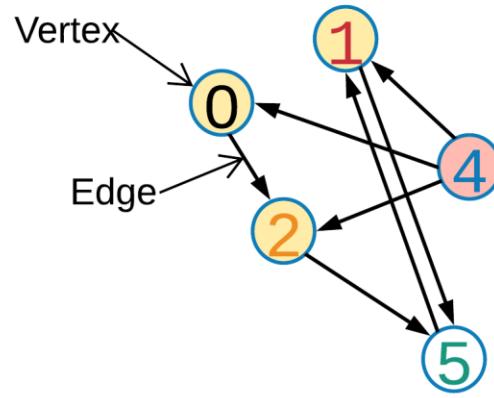


parent	-
Source	-
	0 1 2 3 4 5
parent	- - - - - -
frontier	[]
next	[]



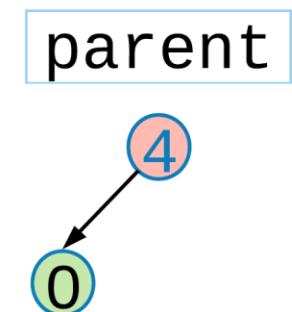
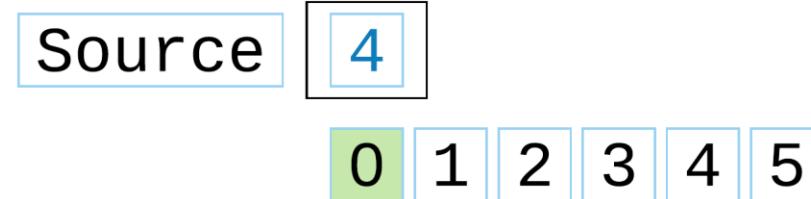
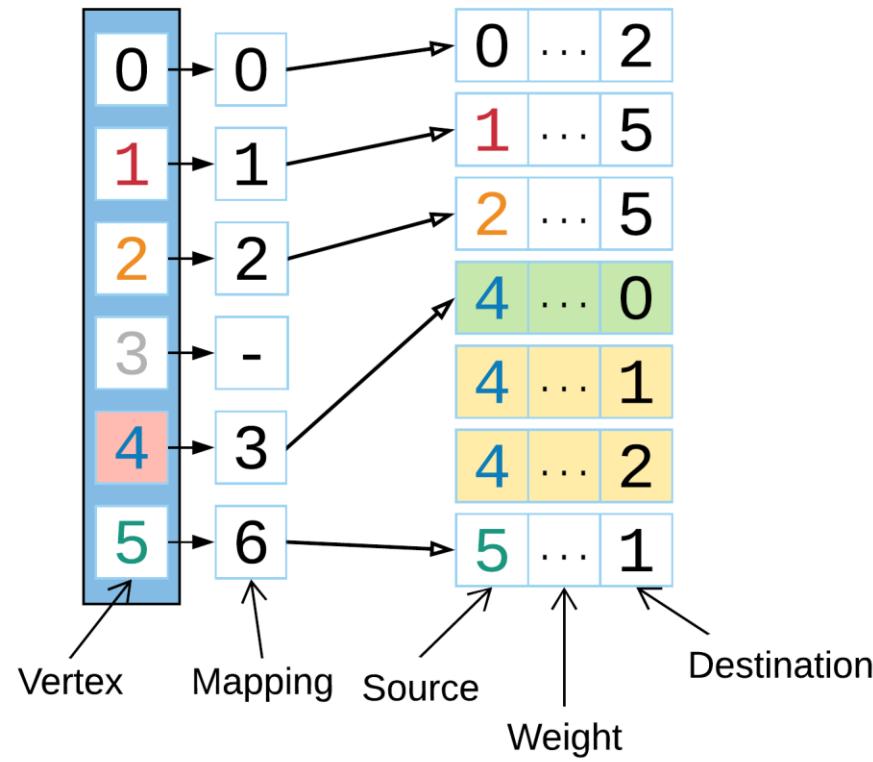
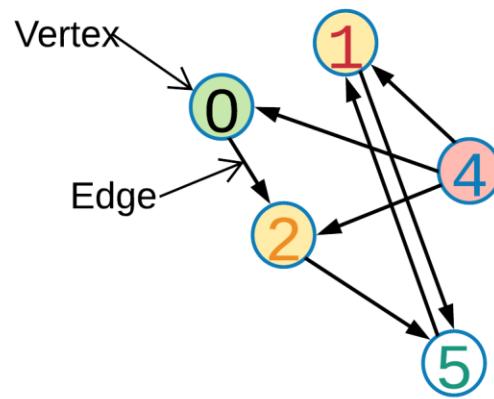
parent

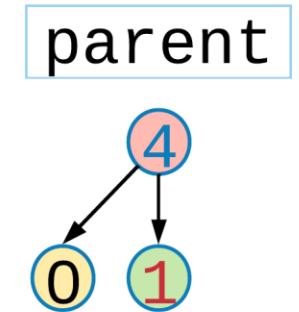
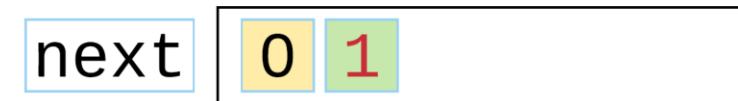
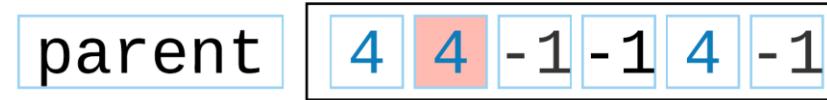
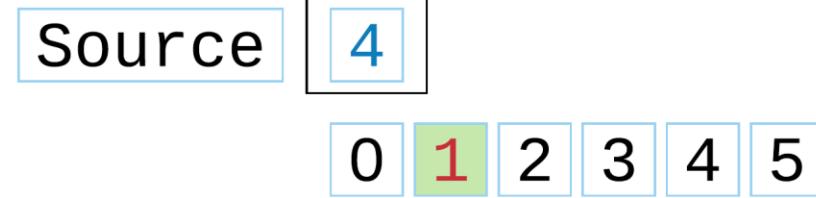
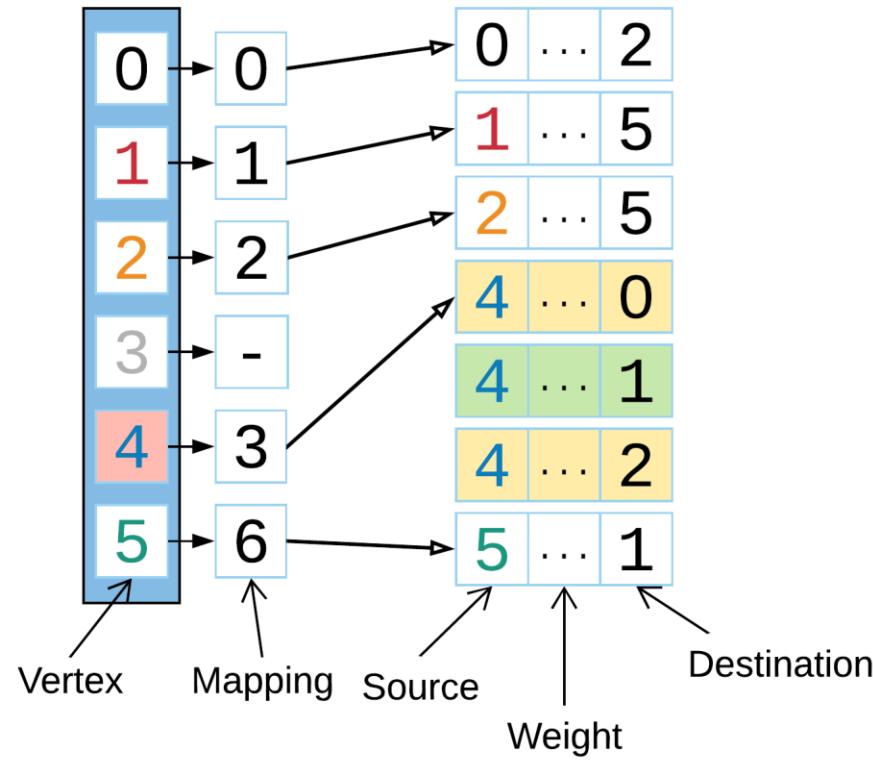
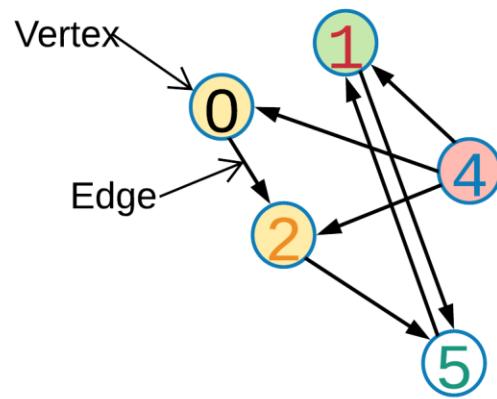
4

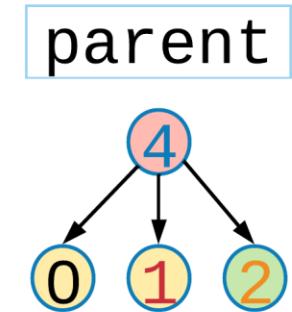
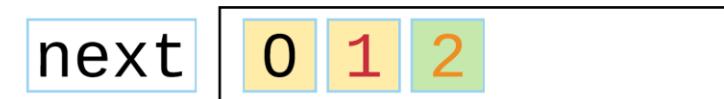
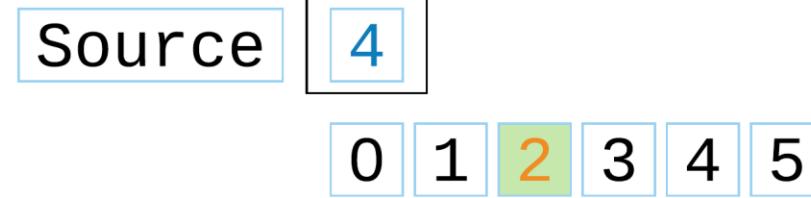
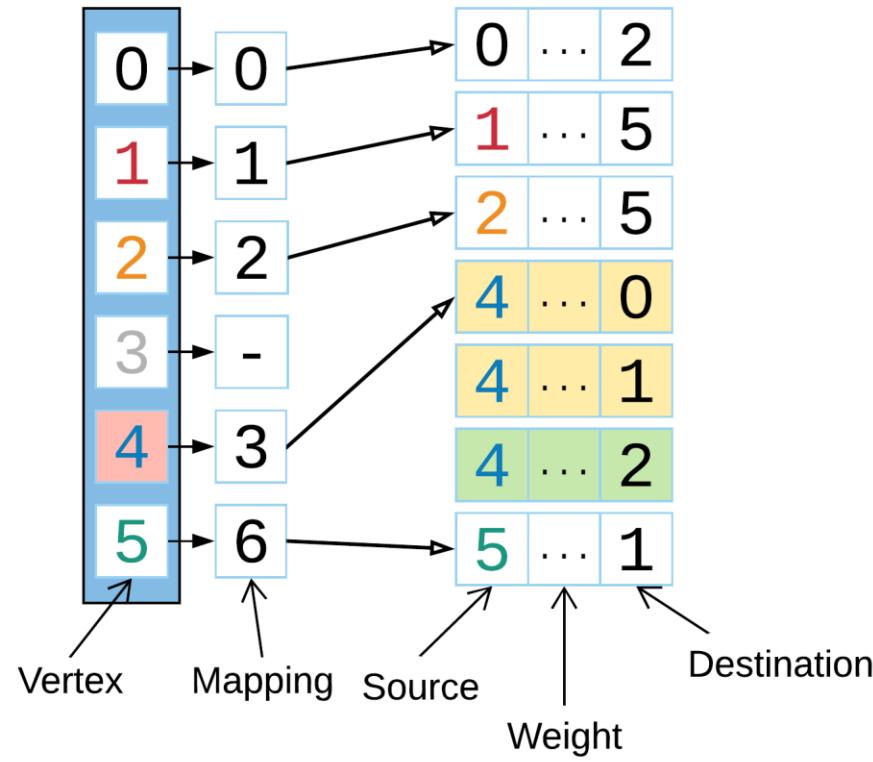
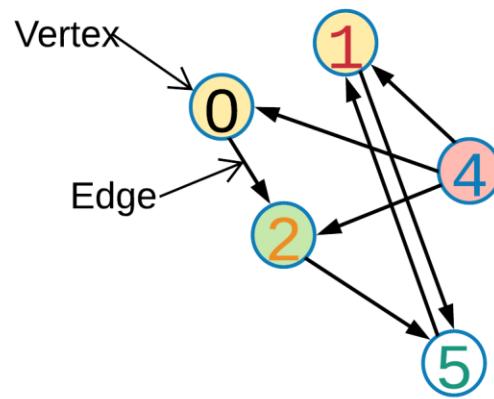


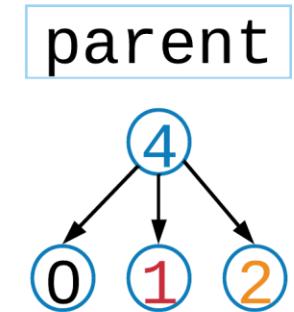
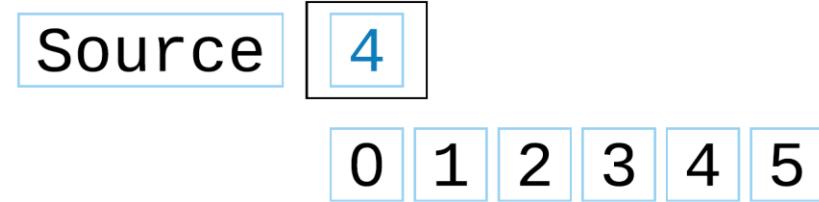
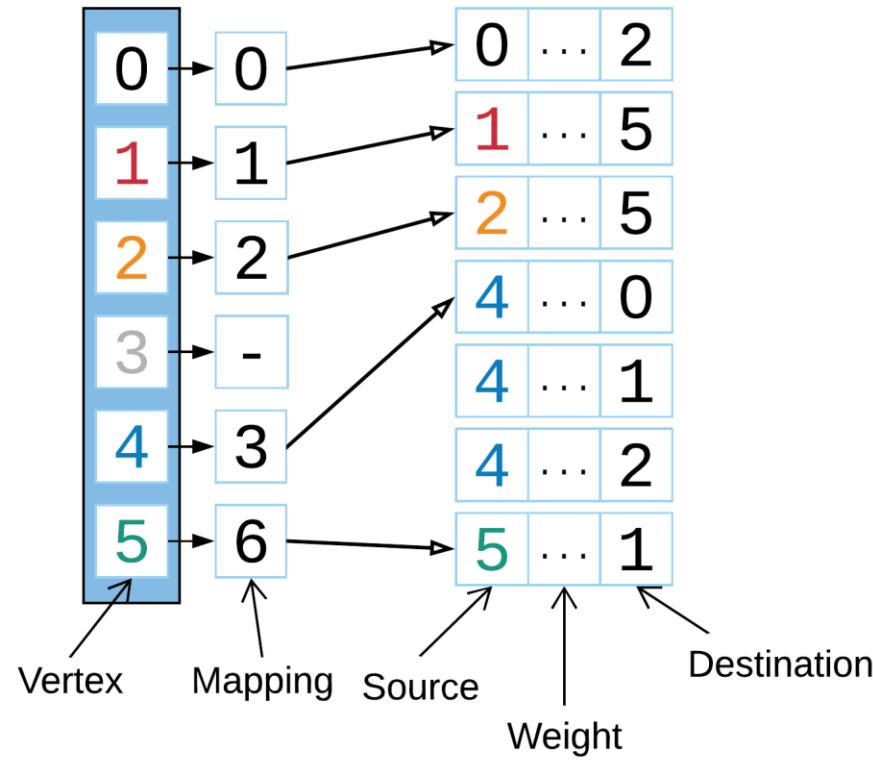
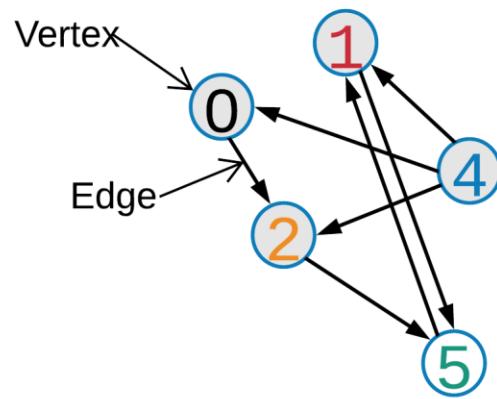
parent

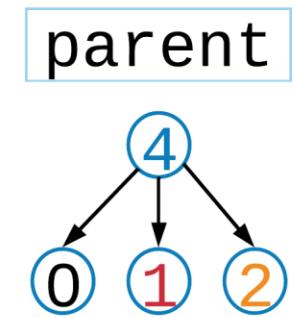
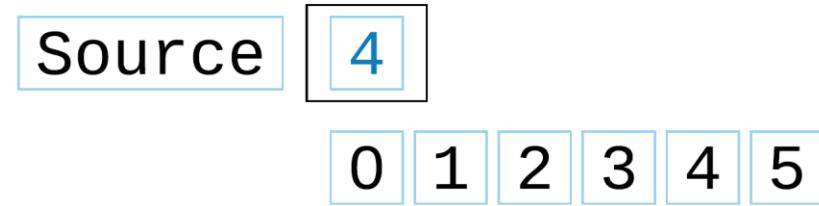
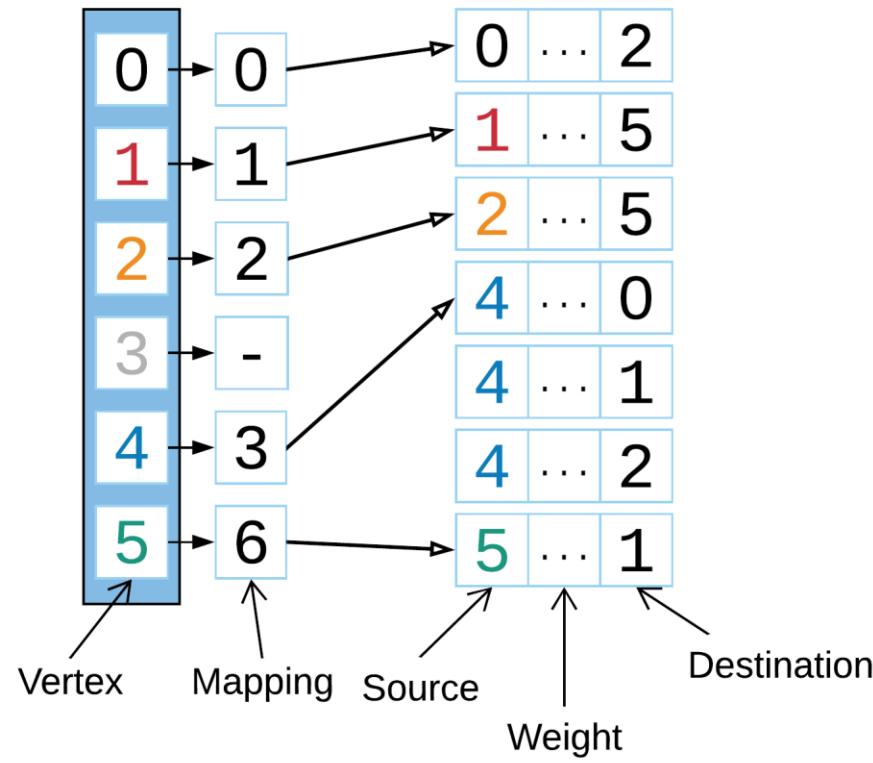
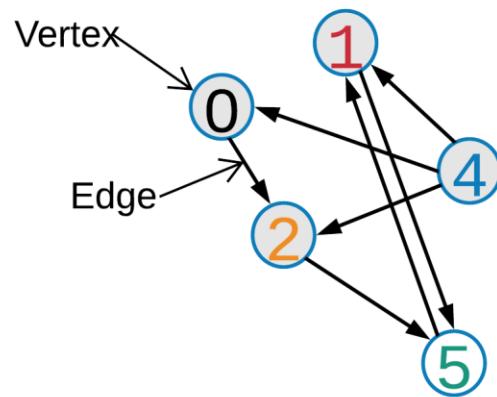
4

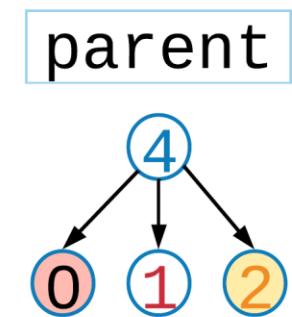
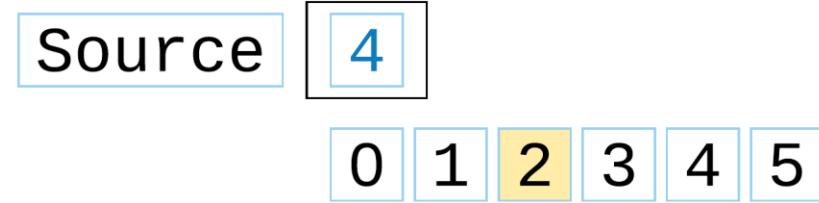
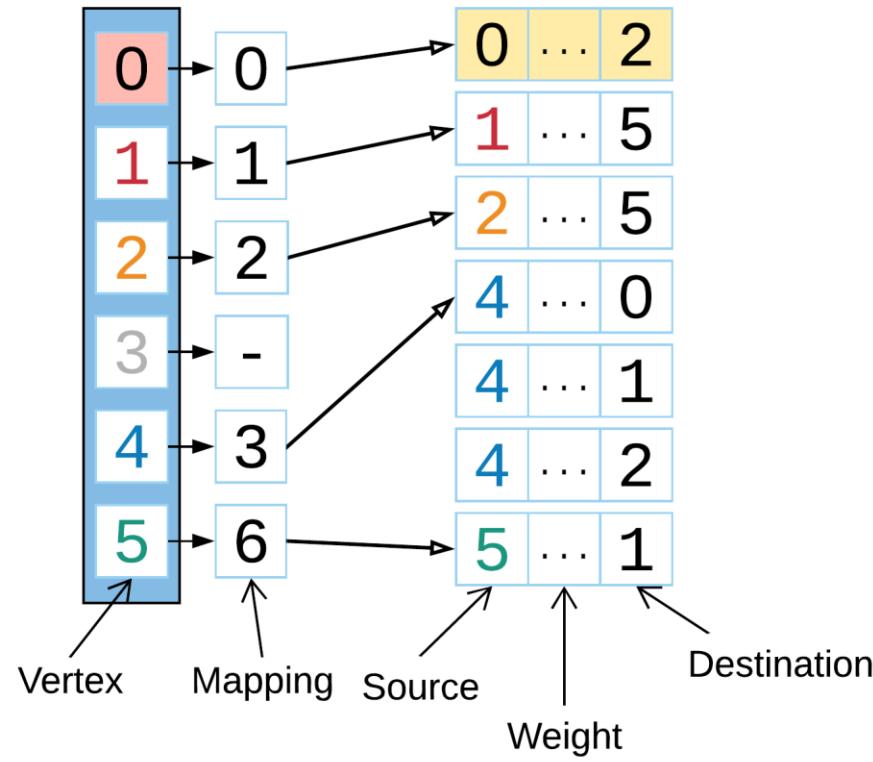
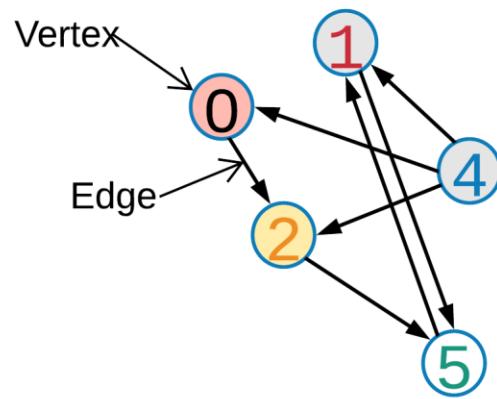


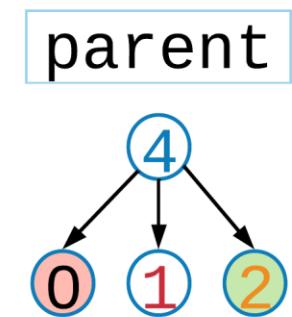
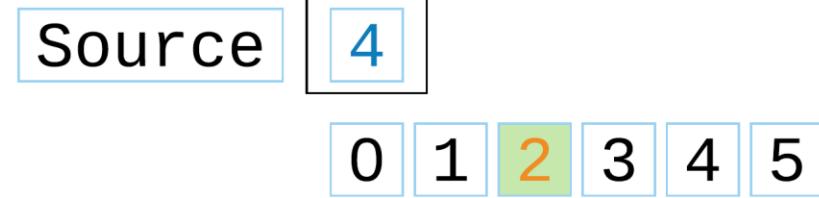
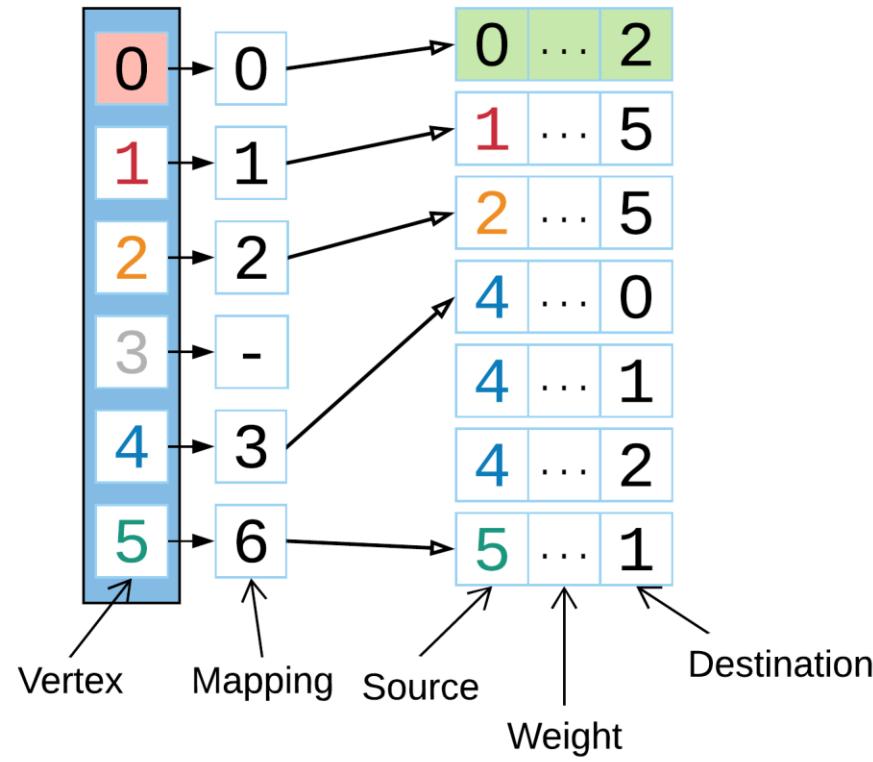
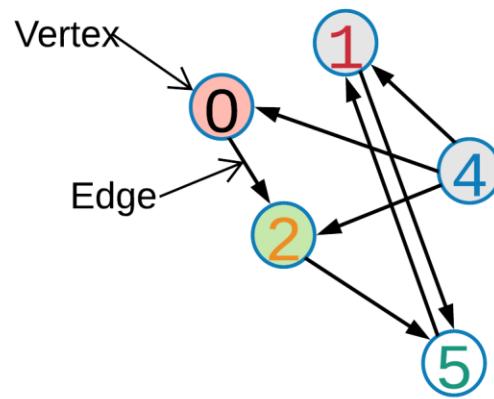


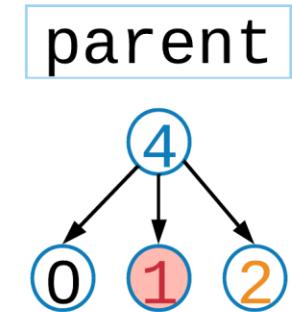
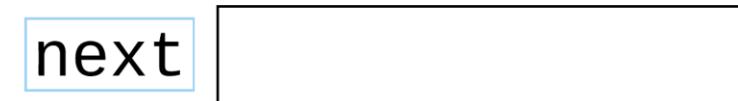
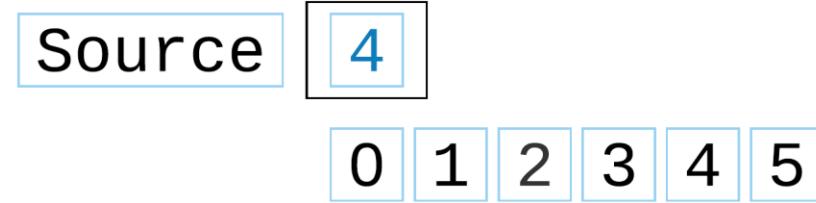
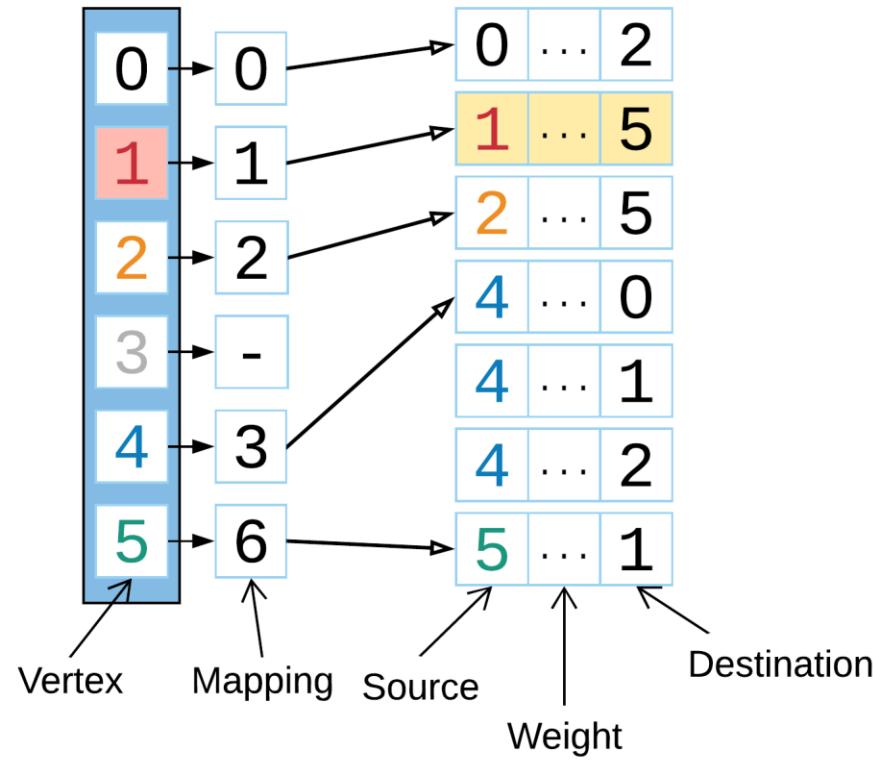
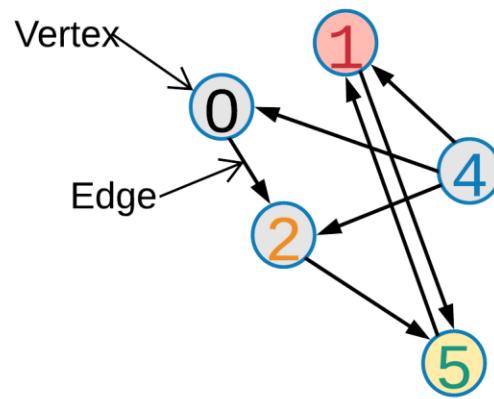


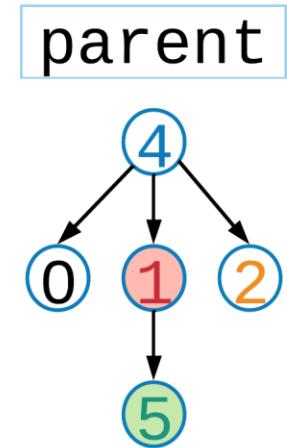
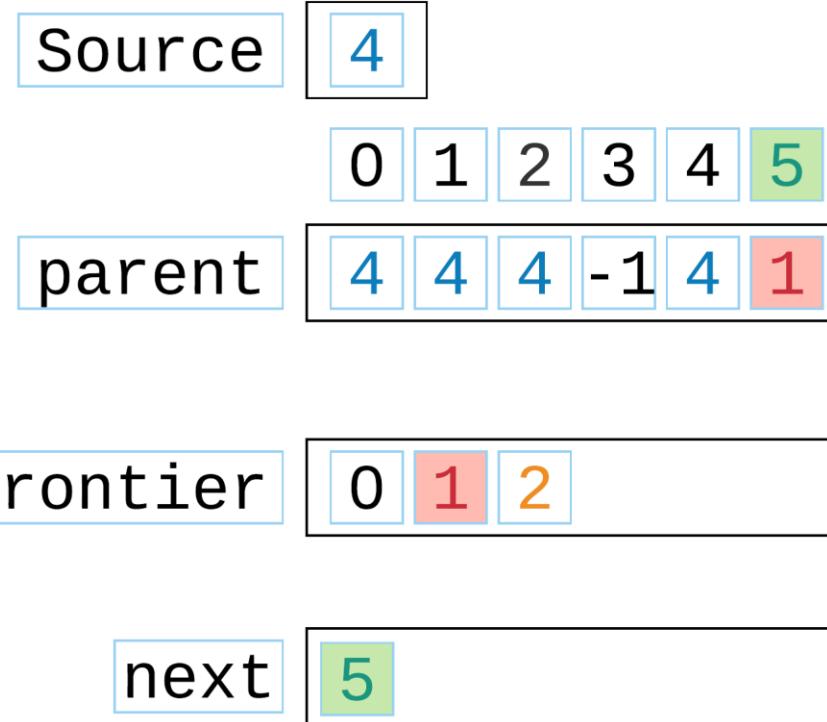
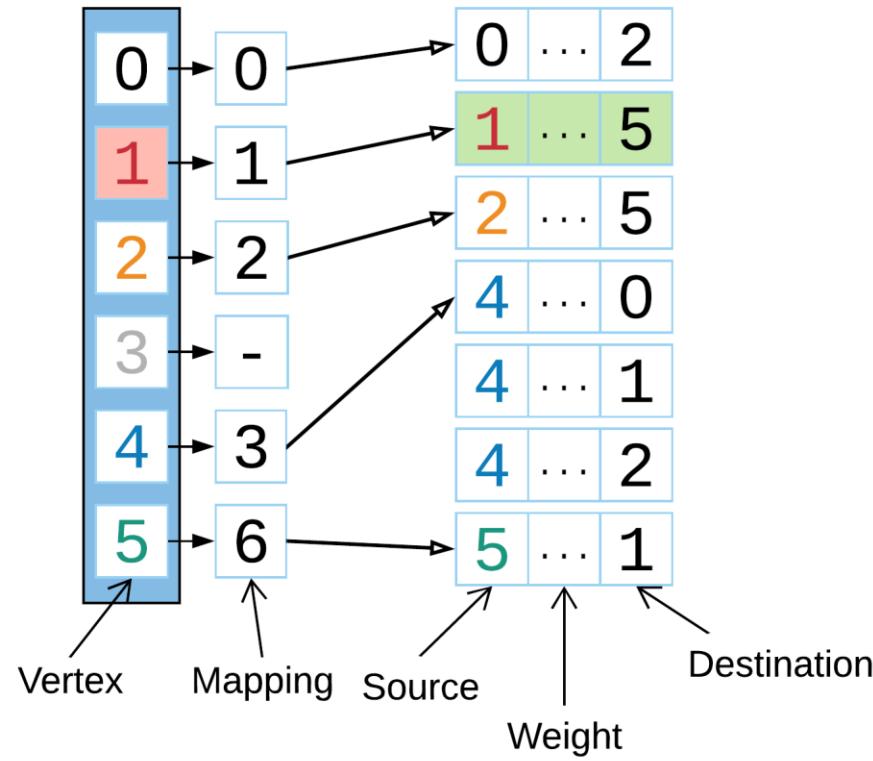
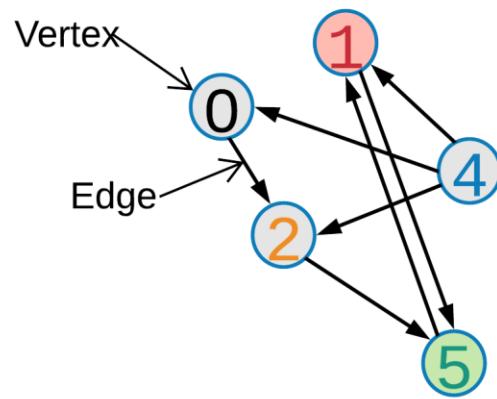


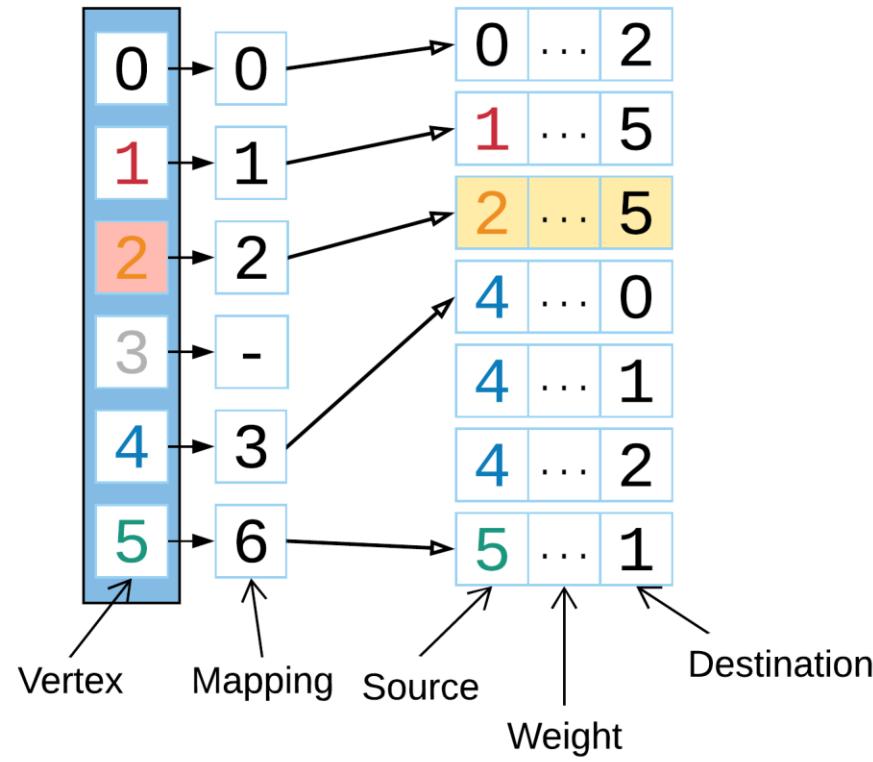
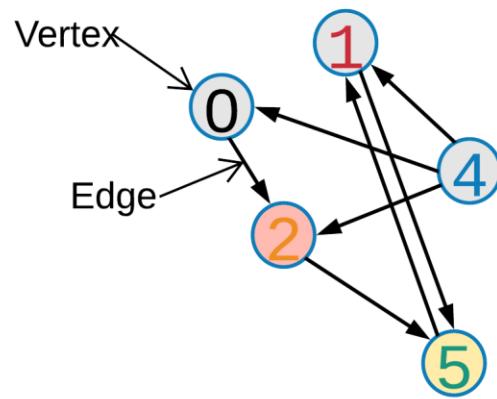




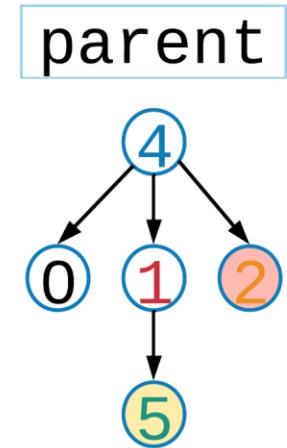


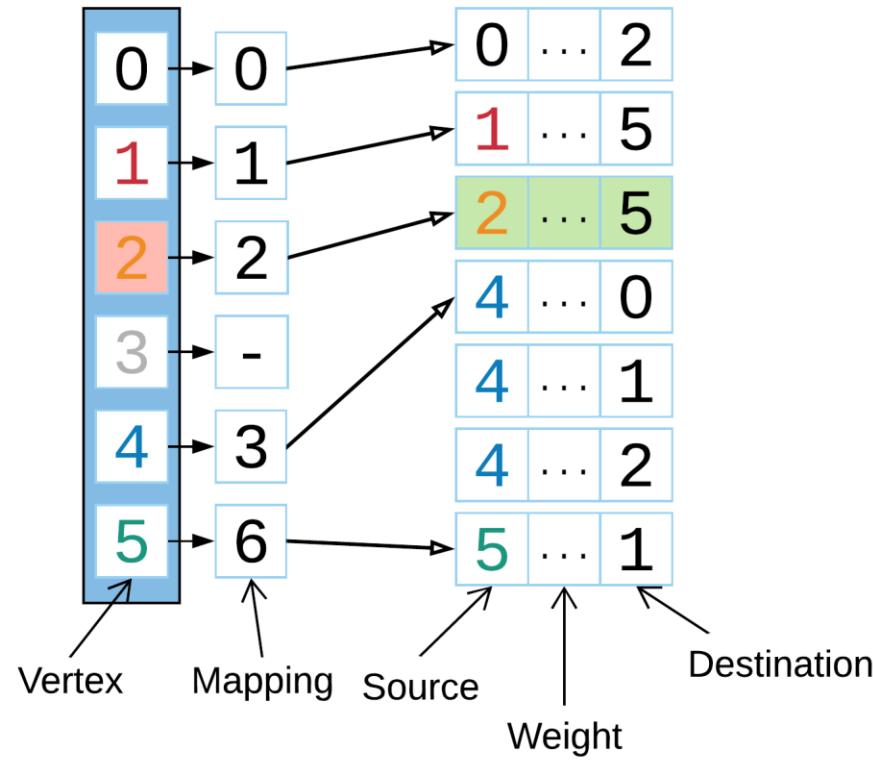
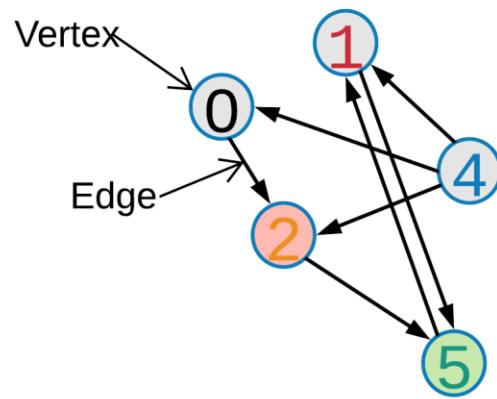




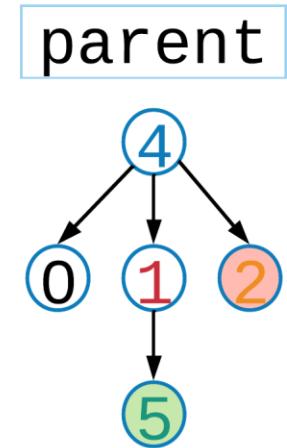


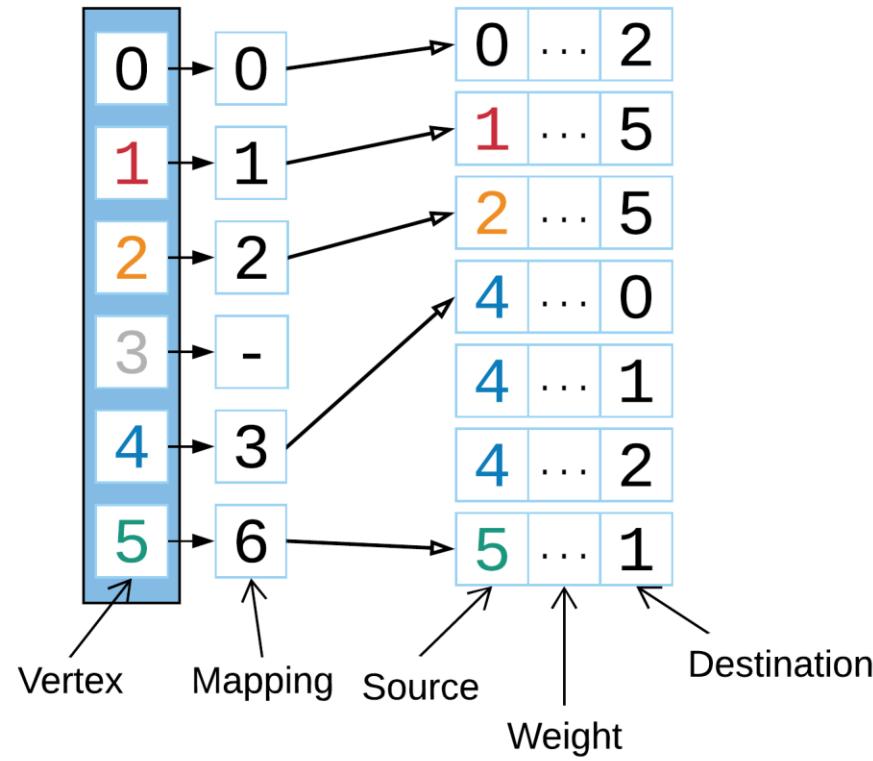
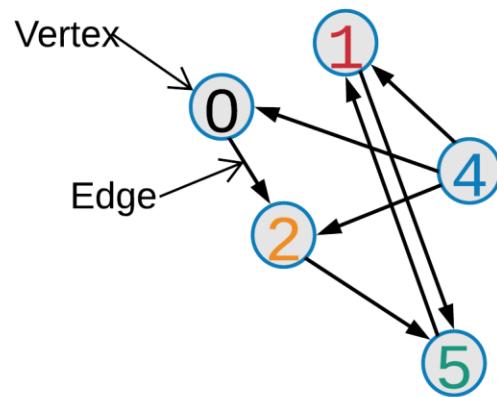
Source	<table border="1"><tr><td>4</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	4	0	1	2	3	4	5
4								
0	1	2	3	4	5			
parent	<table border="1"><tr><td>4</td><td>4</td><td>4</td><td>-1</td><td>4</td><td>1</td></tr></table>	4	4	4	-1	4	1	
4	4	4	-1	4	1			
frontier	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr></table>	0	1	2				
0	1	2						
next	<table border="1"><tr><td>5</td></tr></table>	5						
5								



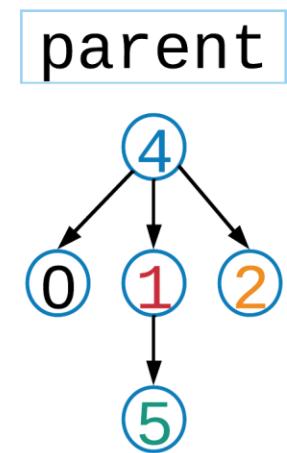


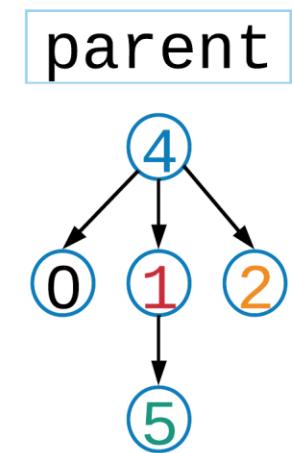
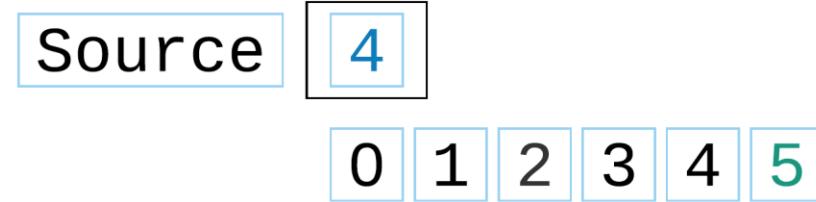
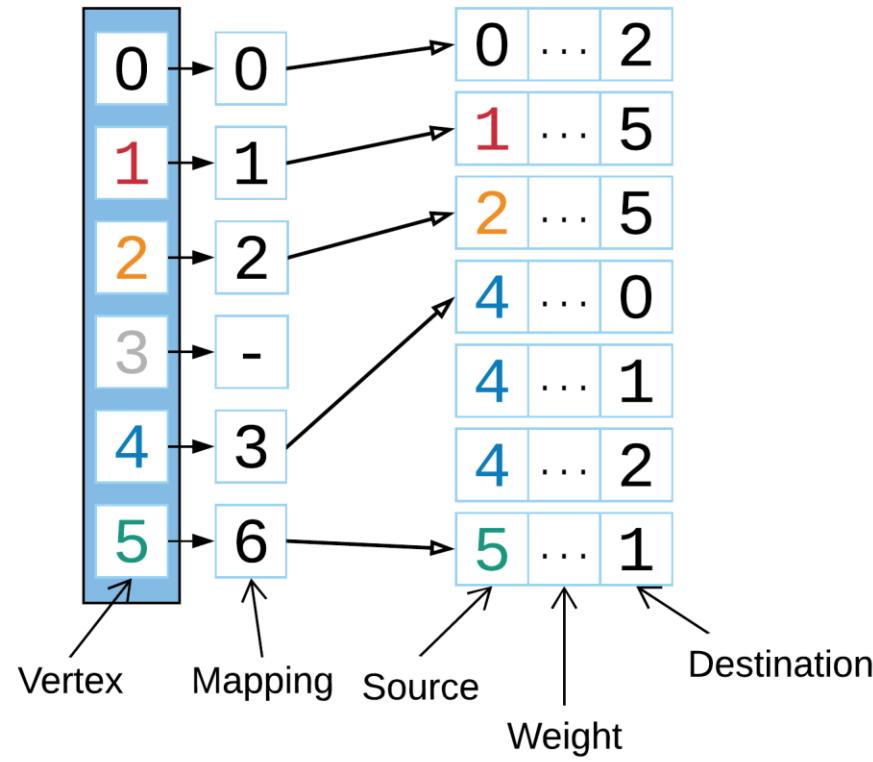
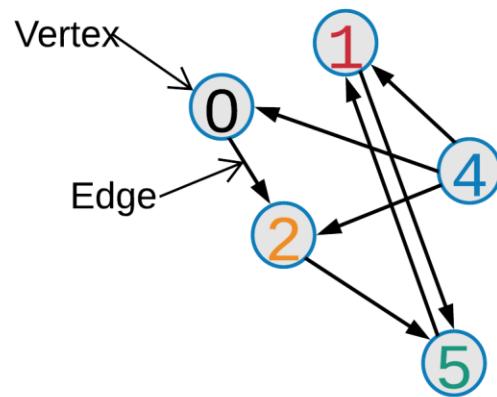
Source	<table border="1"><tr><td>4</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	4	0	1	2	3	4	5
4								
0	1	2	3	4	5			
parent	<table border="1"><tr><td>4</td><td>4</td><td>4</td><td>-1</td><td>4</td><td>1</td></tr></table>	4	4	4	-1	4	1	
4	4	4	-1	4	1			
frontier	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr></table>	0	1	2				
0	1	2						
next	<table border="1"><tr><td>5</td></tr></table>	5						
5								

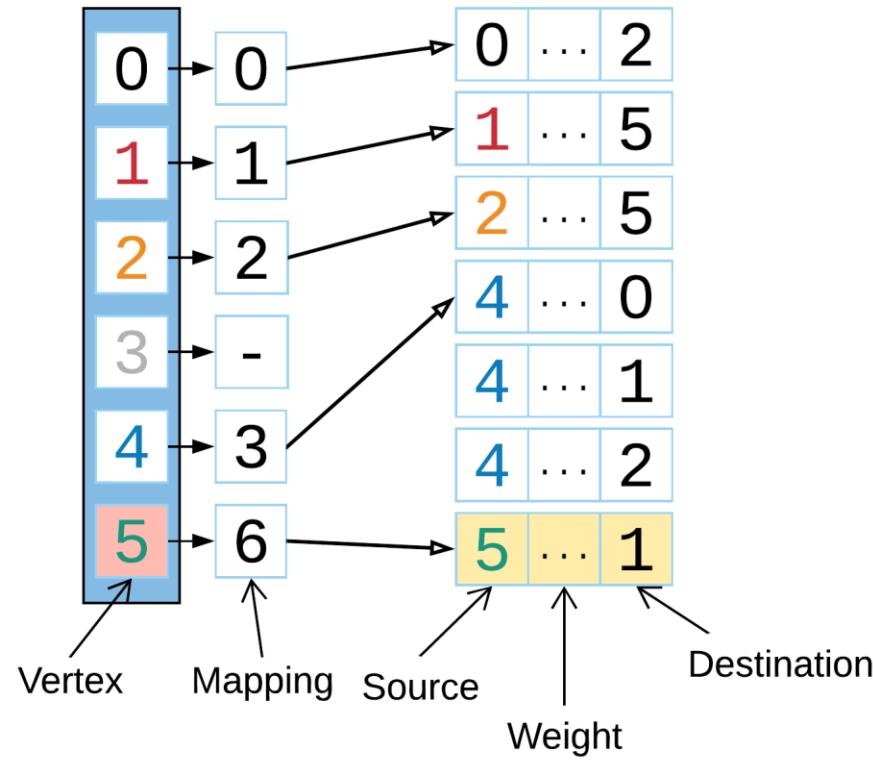
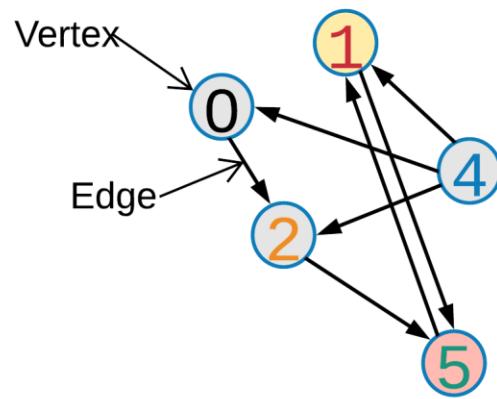




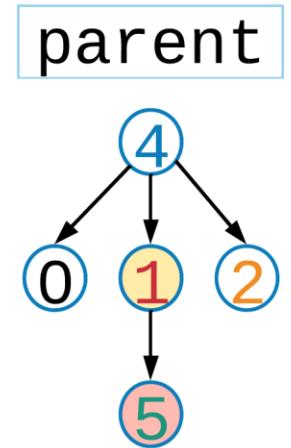
Source	<table border="1"> <tr><td>4</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	4	0	1	2	3	4	5
4								
0	1	2	3	4	5			
parent	<table border="1"> <tr><td>4</td><td>4</td><td>4</td><td>-1</td><td>4</td><td>1</td></tr> </table>	4	4	4	-1	4	1	
4	4	4	-1	4	1			
frontier	<table border="1"> <tr><td>0</td><td>1</td><td>2</td></tr> </table>	0	1	2				
0	1	2						
next	<table border="1"> <tr><td>5</td></tr> </table>	5						
5								

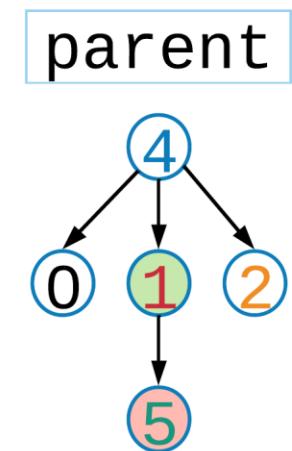
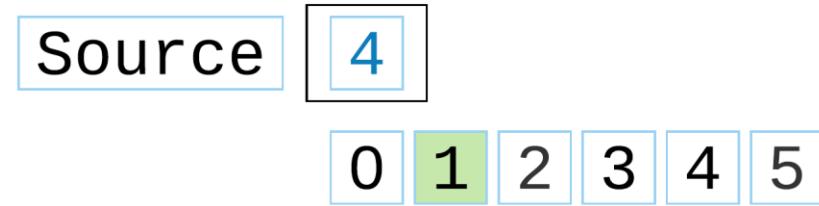
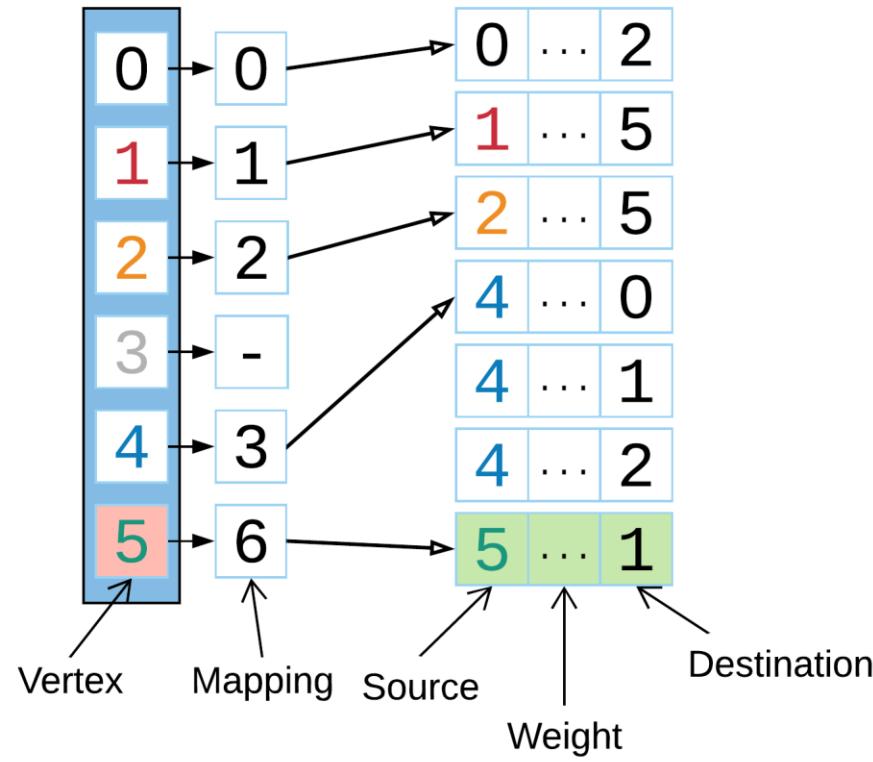
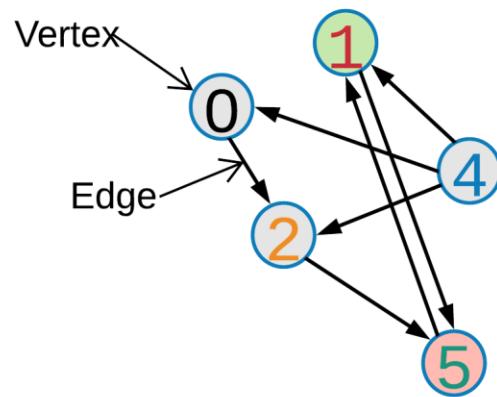


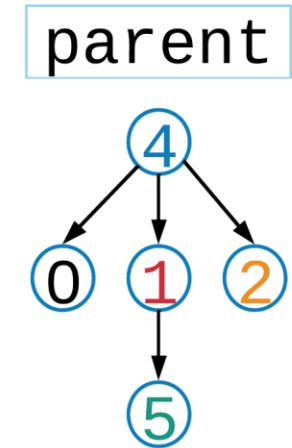
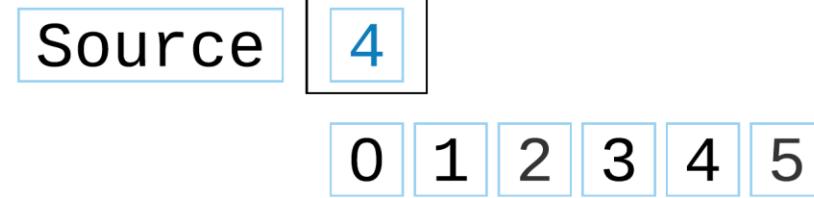
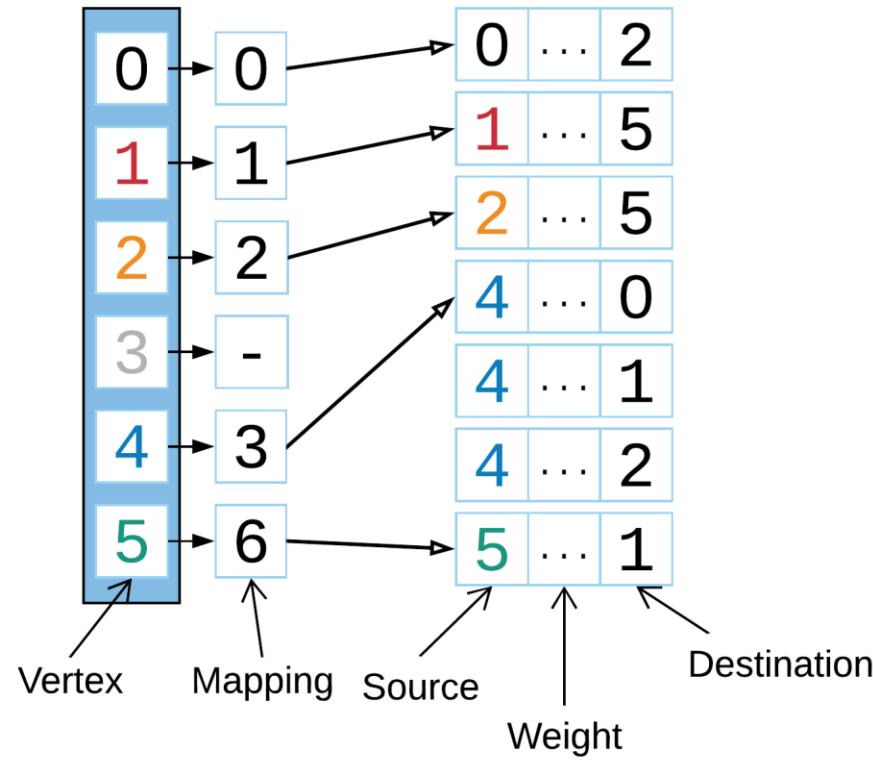
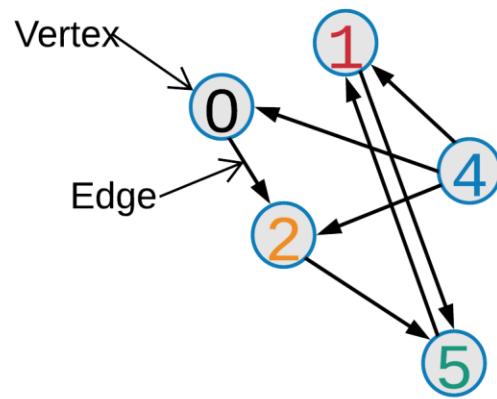




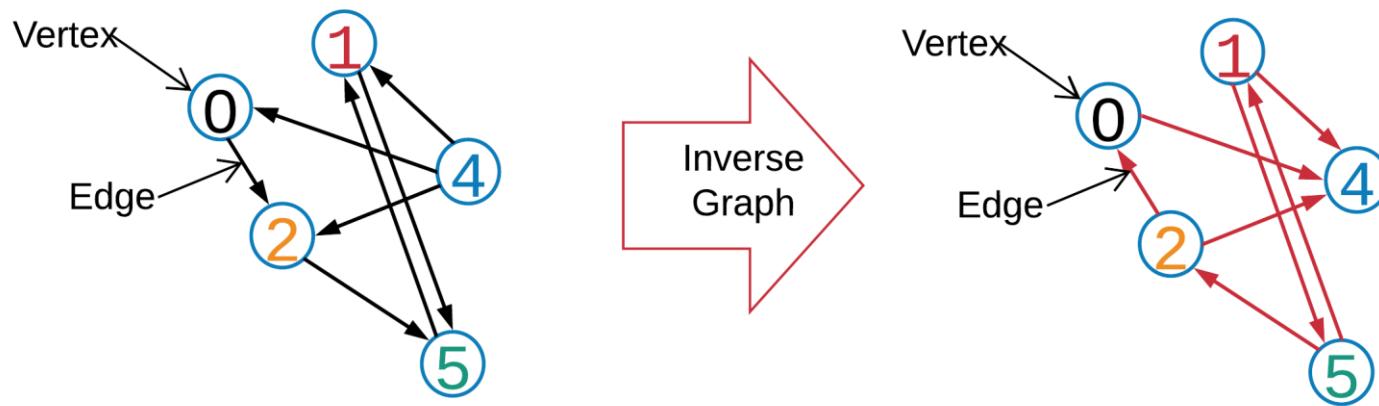
Source	4	0 1 2 3 4 5
parent	4 4 4 -1 4 1	
frontier	5	
next		



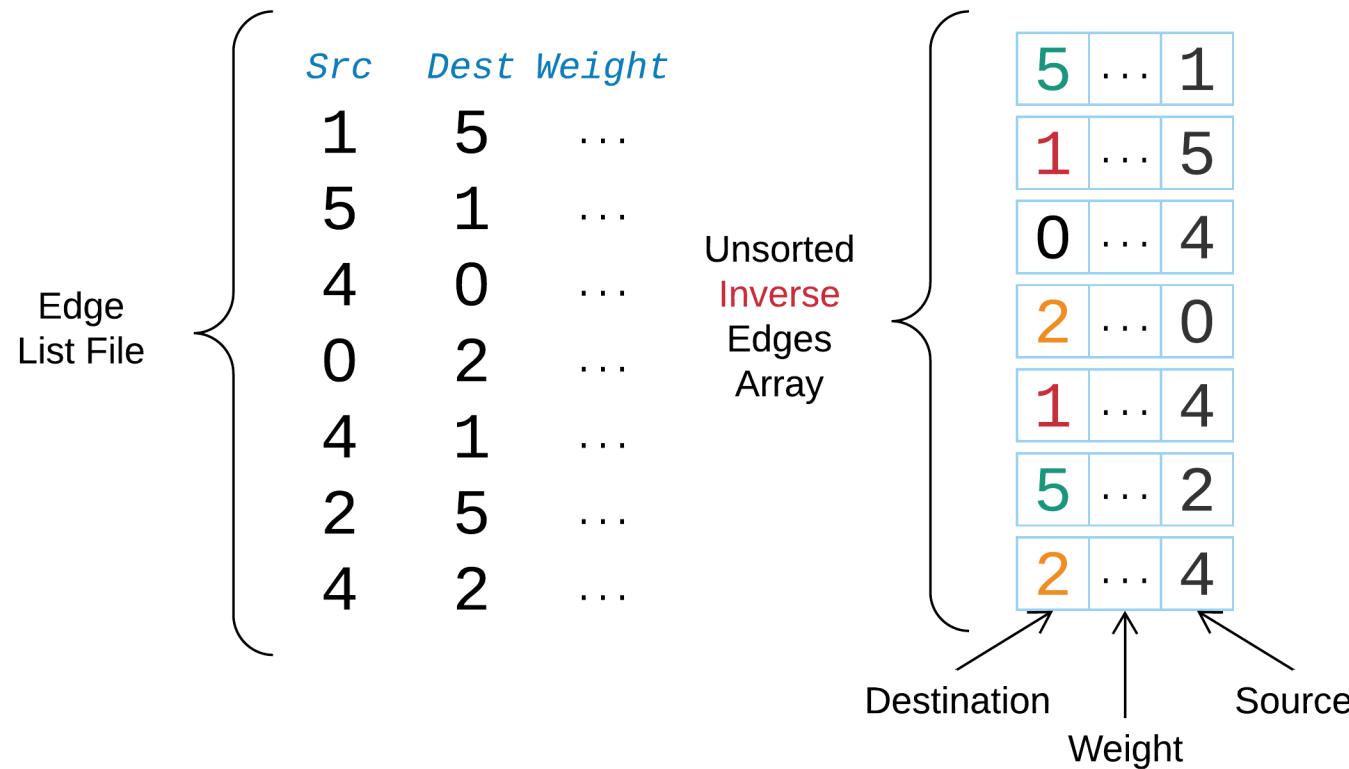




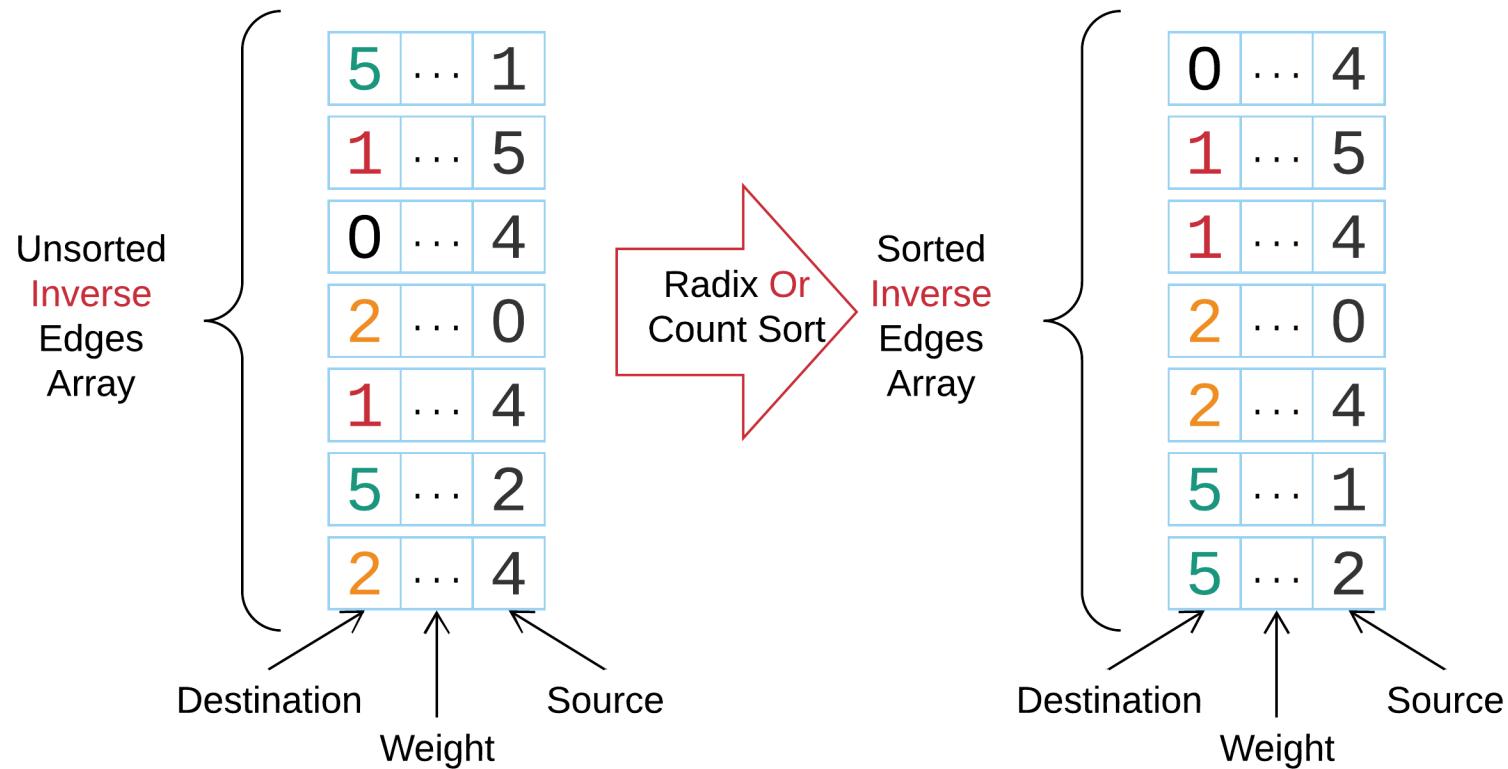
Inverse graphs



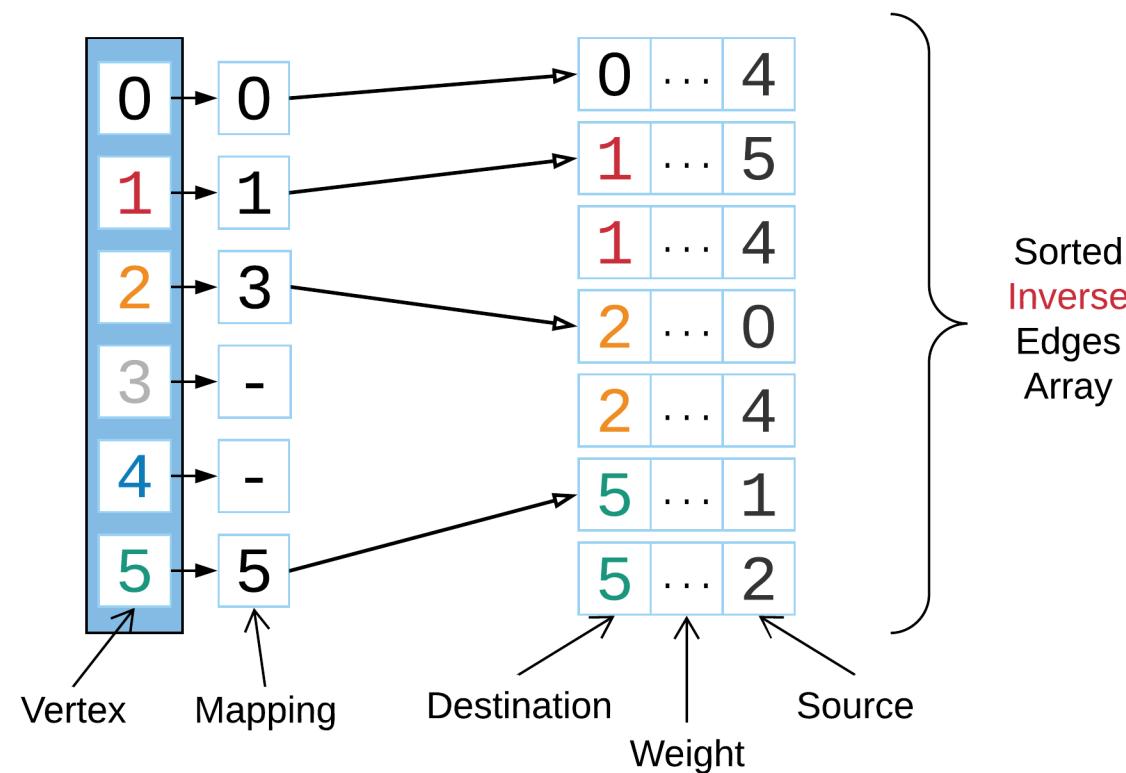
Load src and dest inverted



Sort

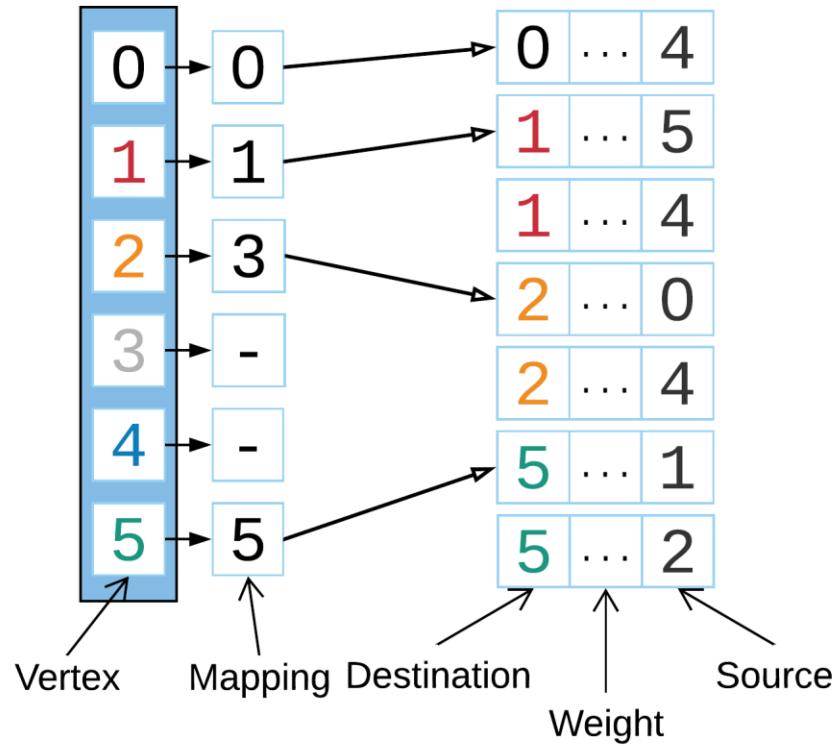
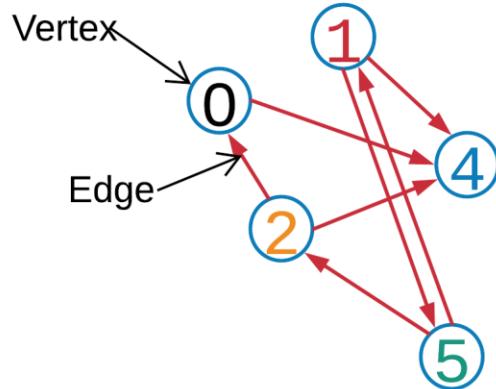


Map vertices



BFS Bottom-Up (Pull)

```
function bottom-up-step(vertices, frontier, next, parents)
    for v ∈ vertices do
        if parents[v] = -1 then
            for n ∈ neighbors[v] do
                if n ∈ frontier do
                    parents[v] ← n
                    next ← next ∪ {v}
                    break
                end if
            end for
        end if
    end for
```



Source

-
0
1
2
3
4
5

parent

-
-
-
-
-
-

frontier

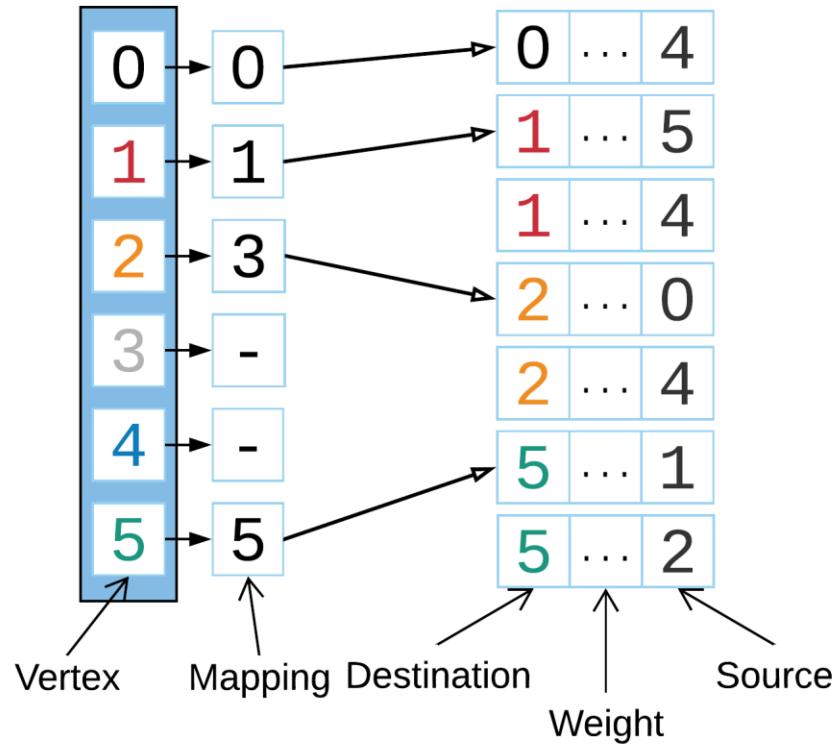
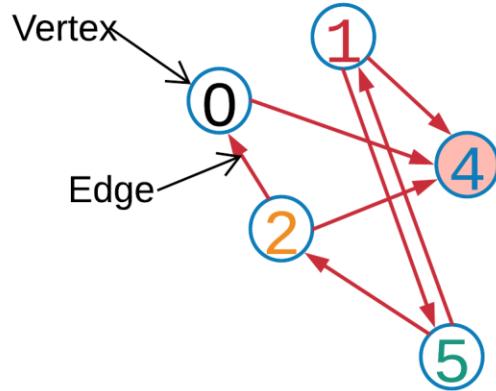
0
0
0
0
0
0

next

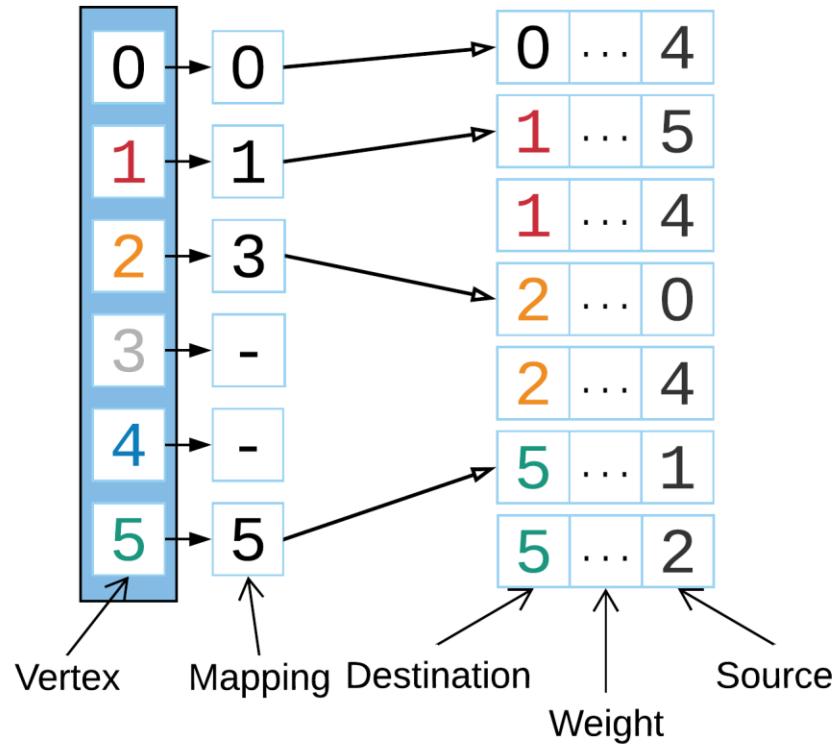
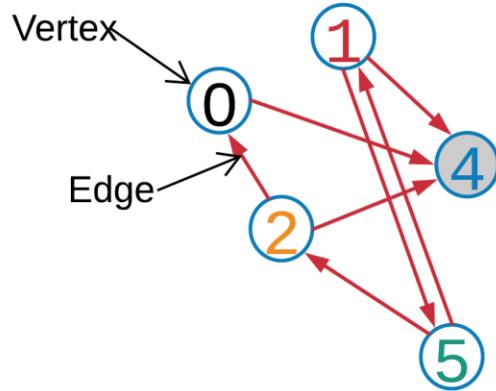
0
0
0
0
0
0

parent

-



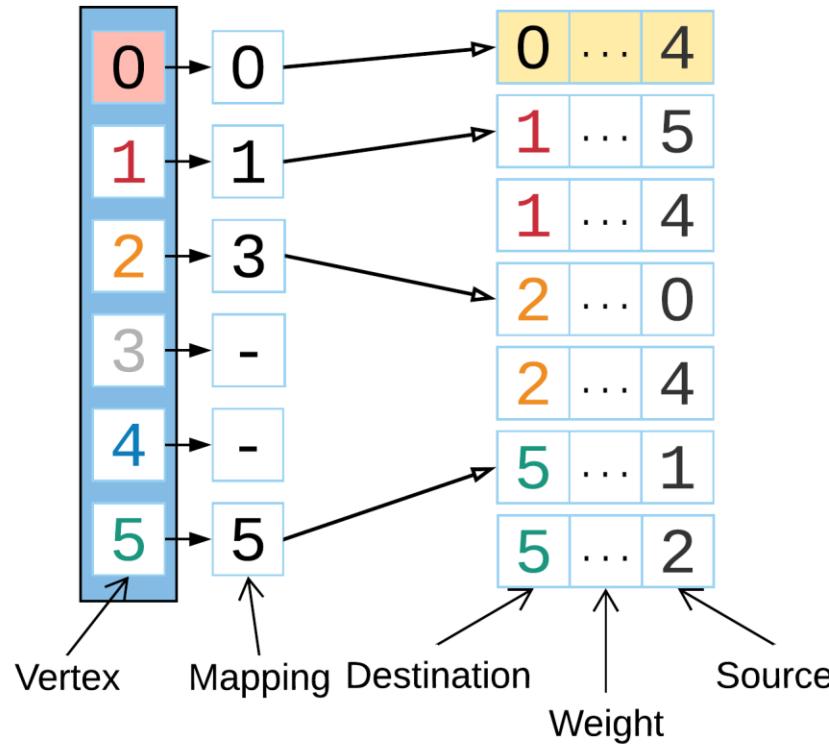
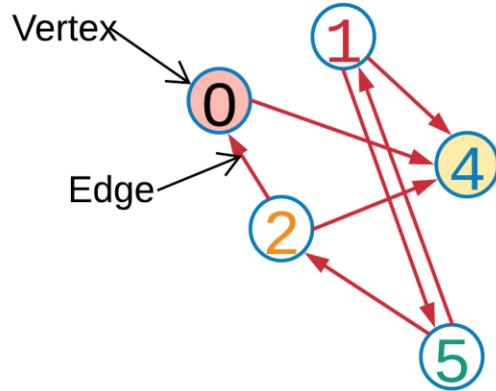
Source	<table border="1"> <tr><td>4</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	4	0	1	2	3	4	5
4								
0	1	2	3	4	5			
parent	<table border="1"> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>4</td><td>-1</td></tr> </table>	-1	-1	-1	-1	4	-1	
-1	-1	-1	-1	4	-1			
frontier	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> </table>	0	0	0	0	1	0	
0	0	0	0	1	0			
next	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	
0	0	0	0	0	0			



Source	<table border="1"> <tr><td>4</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	4	0	1	2	3	4	5
4								
0	1	2	3	4	5			
parent	<table border="1"> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>4</td><td>-1</td></tr> </table>	-1	-1	-1	-1	4	-1	
-1	-1	-1	-1	4	-1			
frontier	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> </table>	0	0	0	0	1	0	
0	0	0	0	1	0			
next	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	
0	0	0	0	0	0			

parent

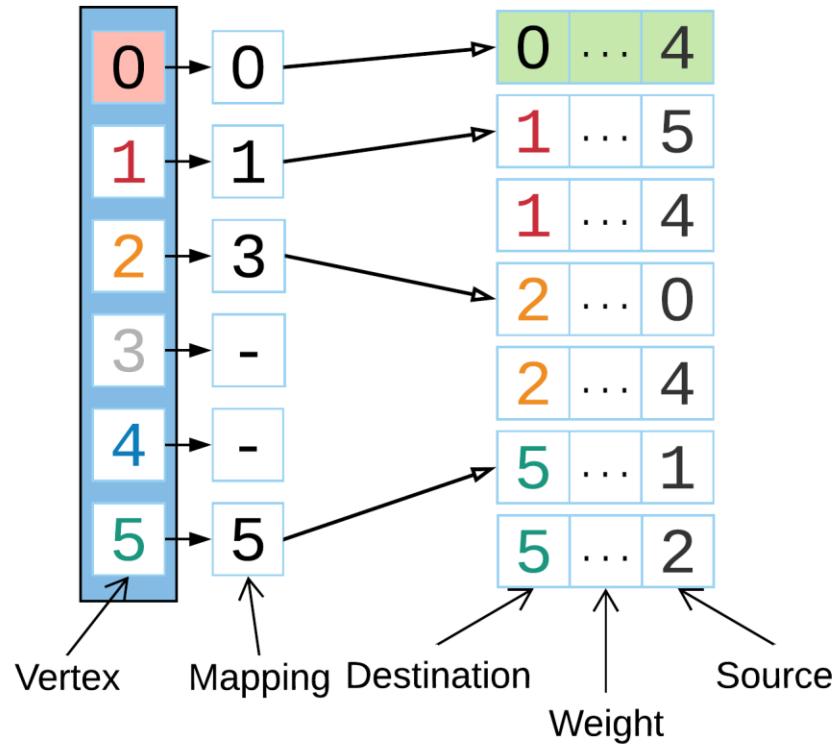
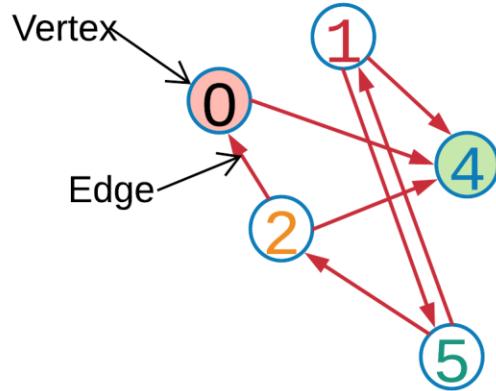
4



Source	<table border="1"> <tr><td>4</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	4	0	1	2	3	4	5
4								
0	1	2	3	4	5			
parent	<table border="1"> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>4</td><td>-1</td></tr> </table>	-1	-1	-1	-1	4	-1	
-1	-1	-1	-1	4	-1			
frontier	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> </table>	0	0	0	0	1	0	
0	0	0	0	1	0			
next	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	
0	0	0	0	0	0			

parent

4



Source

4

0	1	2	3	4	5
---	---	---	---	---	---

parent

4	-1	-1	-1	4	-1
---	----	----	----	---	----

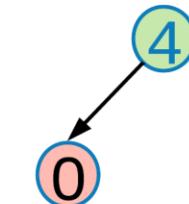
frontier

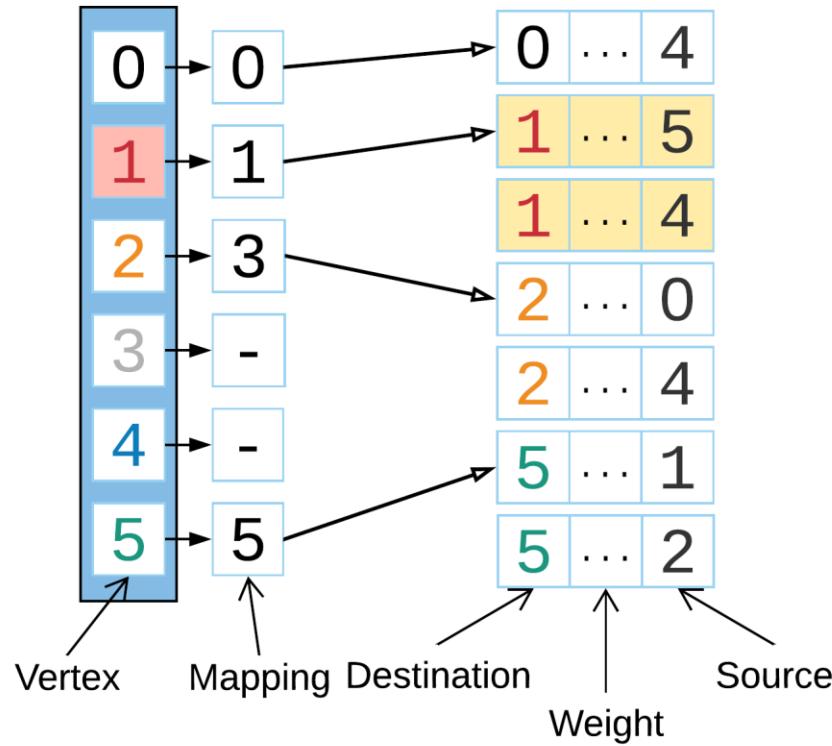
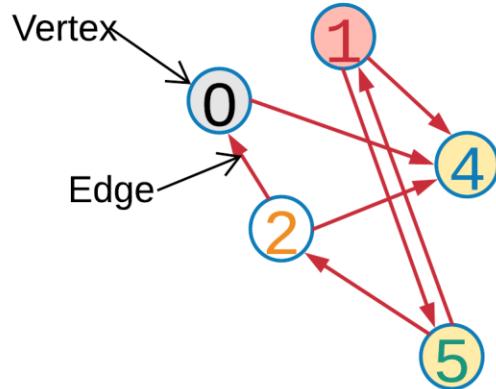
0	0	0	0	1	0
---	---	---	---	---	---

next

1	0	0	0	0	0
---	---	---	---	---	---

parent





Source

4

0 1 2 3 4 5

parent

4 -1 -1 -1 4 -1

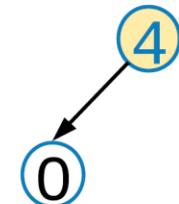
frontier

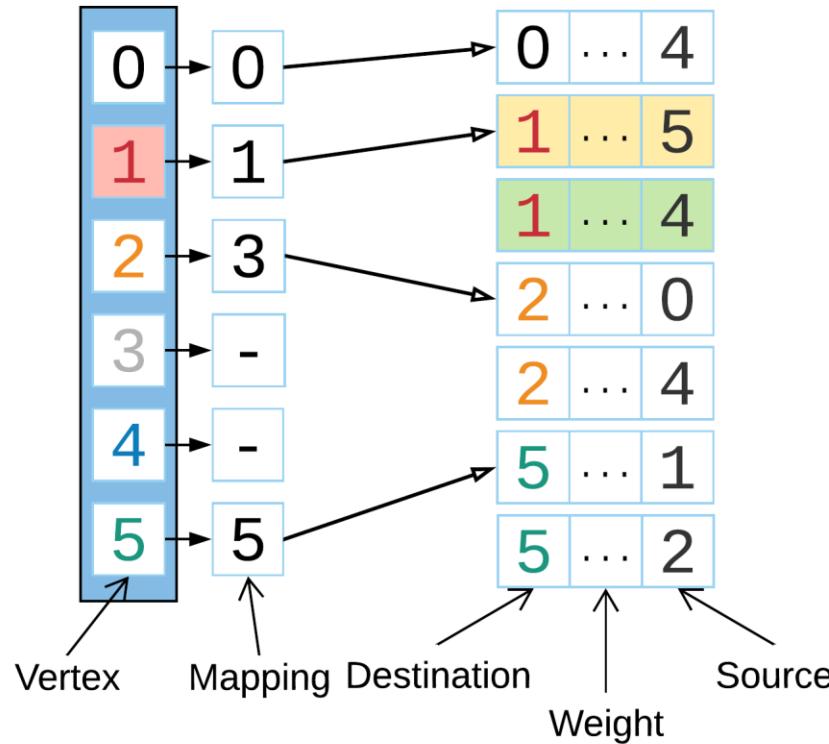
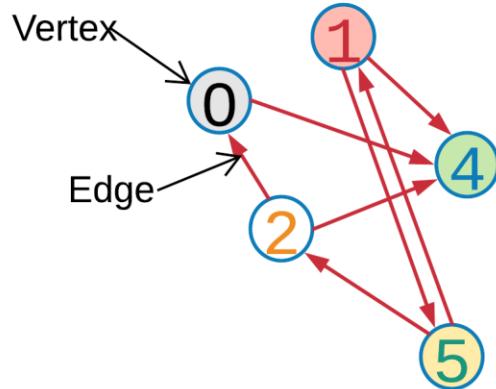
0 0 0 0 1 0

next

1 0 0 0 0 0

parent





Source [4]

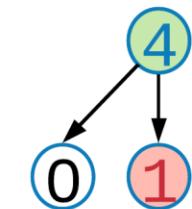
0 1 2 3 4 5

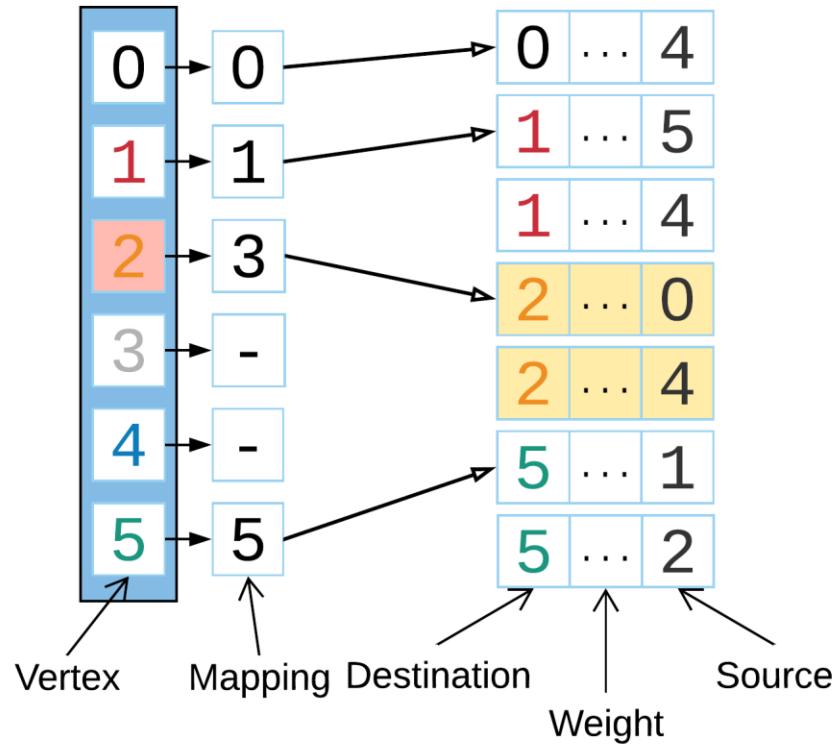
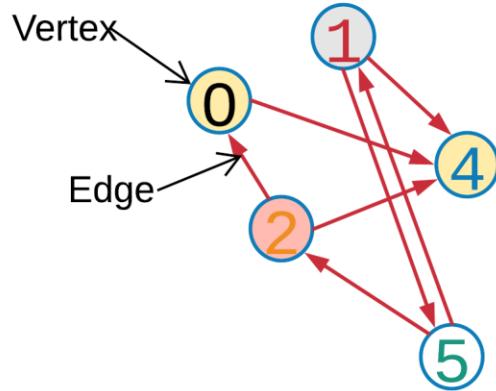
parent [4 4 -1 -1 4 -1]

frontier [0 0 0 0 1 0]

next [1 1 0 0 0 0]

parent





Source [4]

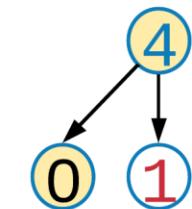
0 1 2 3 4 5

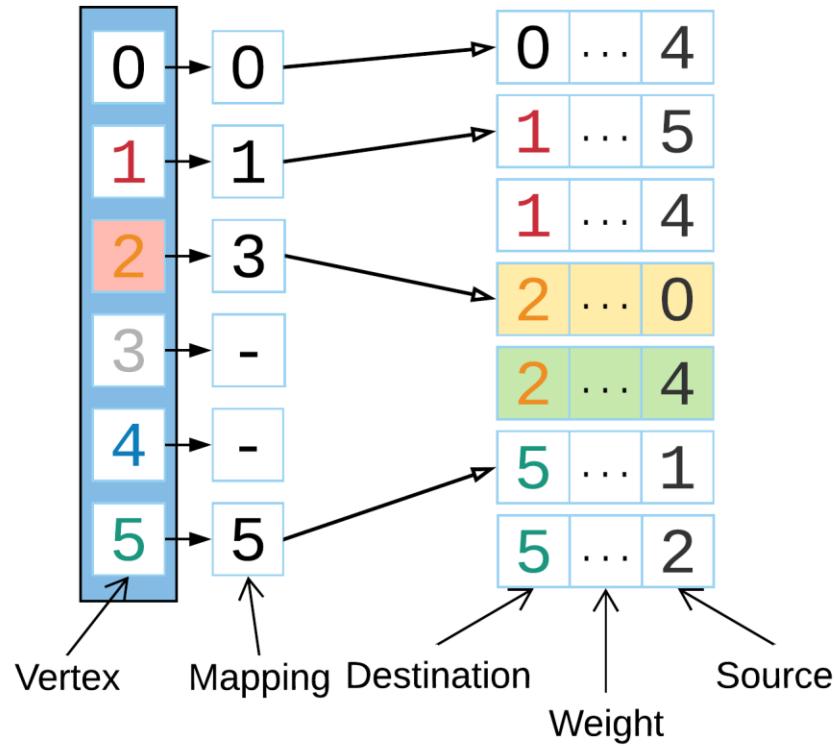
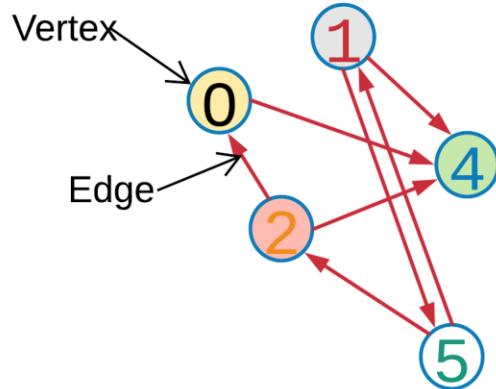
parent [4 4 -1 -1 4 -1]

frontier [0 0 0 0 1 0]

next [1 1 0 0 0 0]

parent





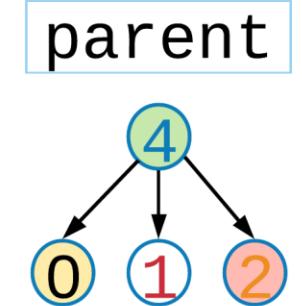
Source [4]

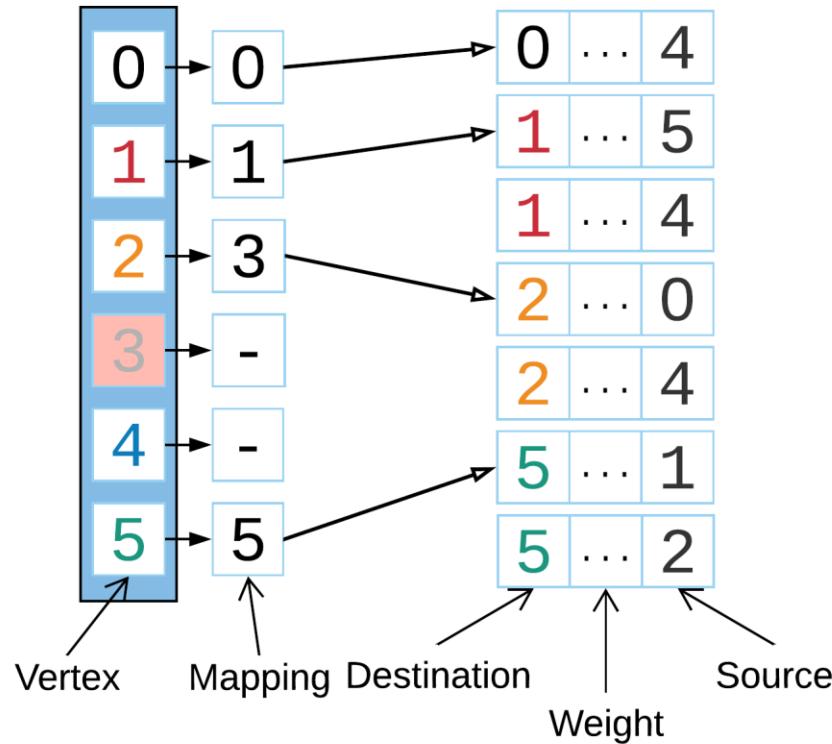
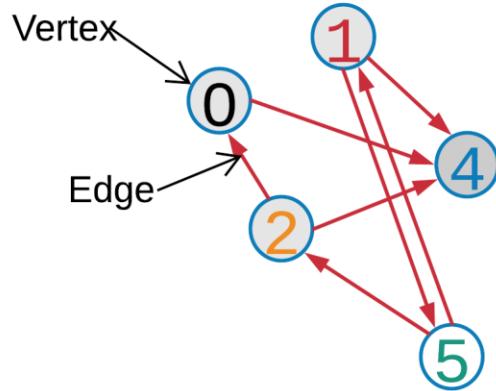
0 1 2 3 4 5

parent [4 4 4 -1 4 -1]

frontier [0 0 0 0 1 0]

next [1 1 1 0 0 0]



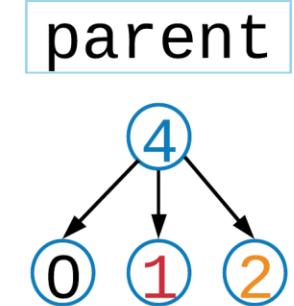


Source

4					
0	1	2	3	4	5

parent

4	4	4	-1	4	-1
---	---	---	----	---	----

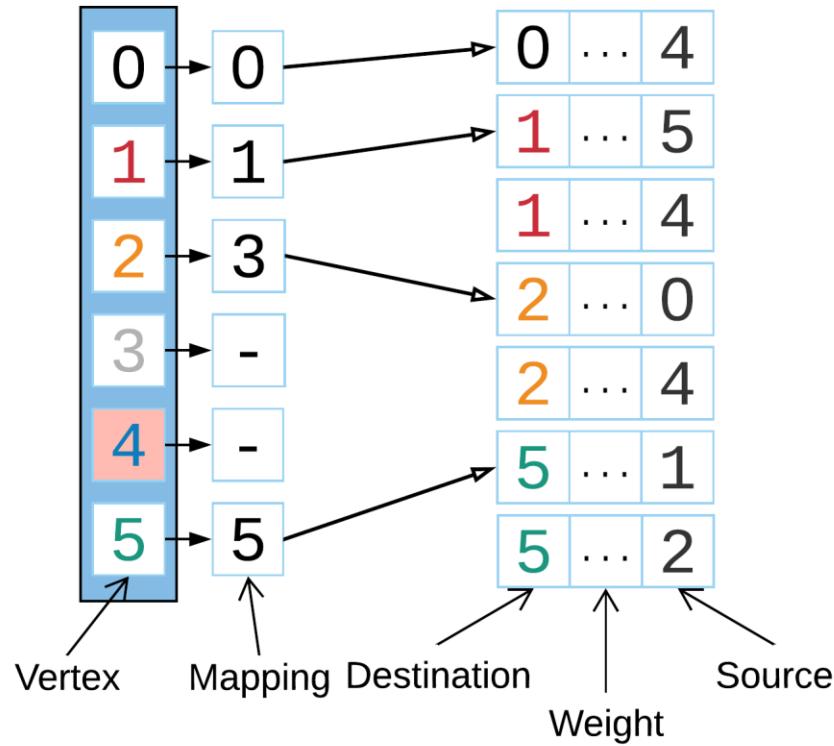
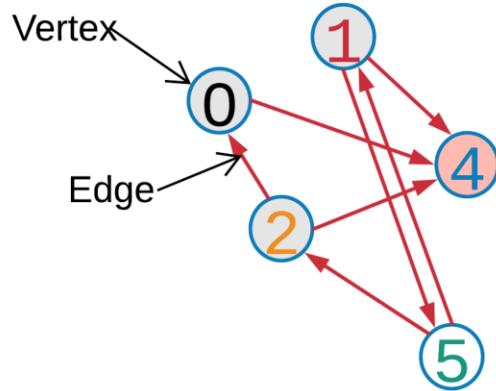


frontier

0	0	0	0	1	0
---	---	---	---	---	---

next

1	1	1	0	0	0
---	---	---	---	---	---



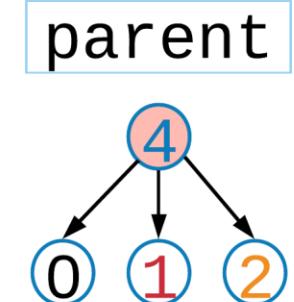
Source [4]

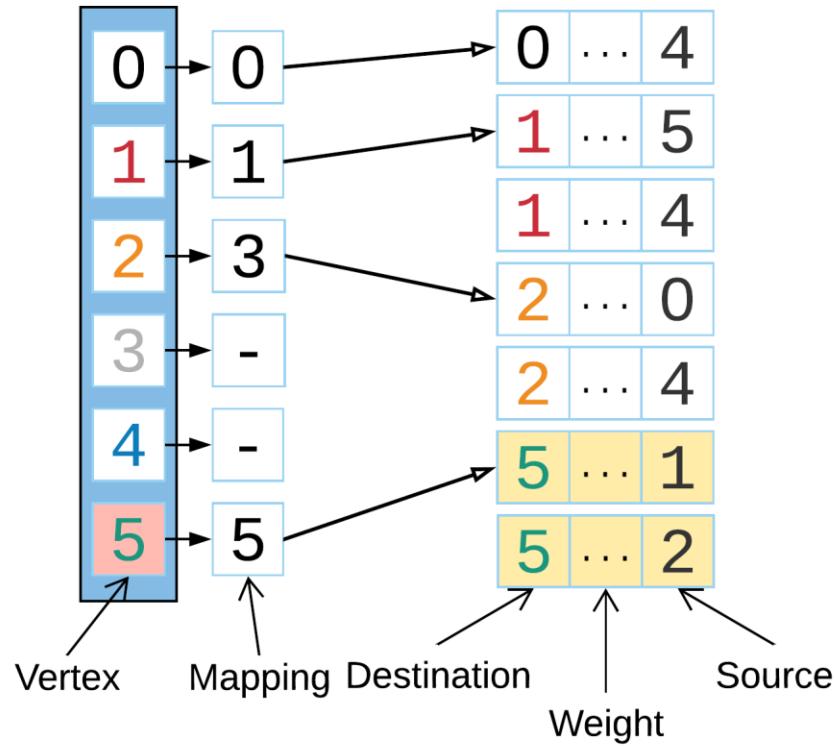
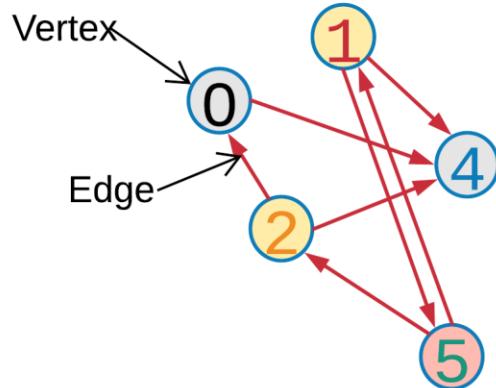
0 1 2 3 4 5

parent [4 4 4 -1 4 -1]

frontier [0 0 0 0 1 0]

next [1 1 1 0 0 0]





Source

4					
0	1	2	3	4	5

parent

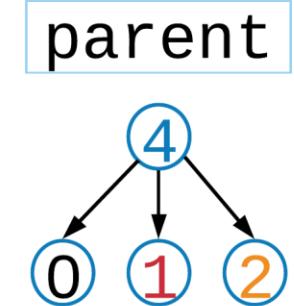
4	4	4	-1	4	-1
---	---	---	----	---	----

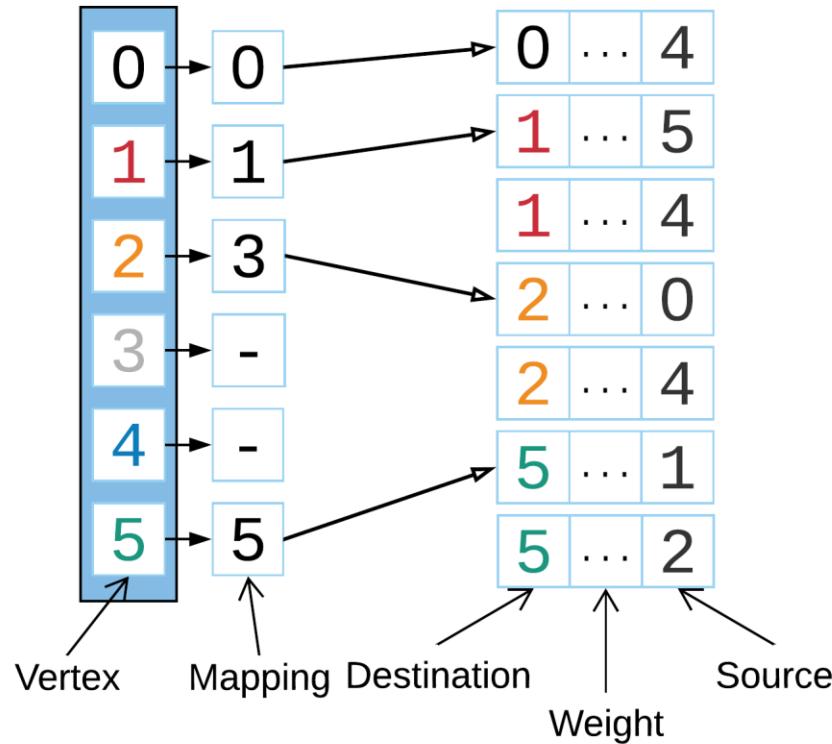
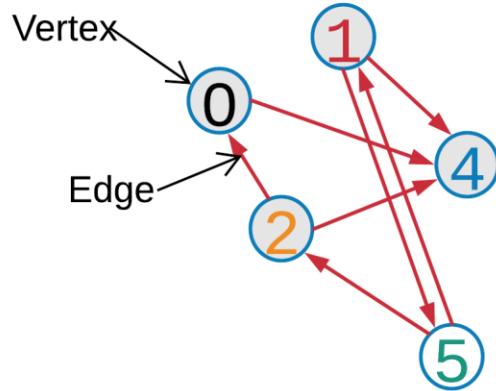
frontier

0	0	0	0	1	0
---	---	---	---	---	---

next

1	1	1	0	0	0
---	---	---	---	---	---





Source

4

0	1	2	3	4	5
---	---	---	---	---	---

parent

4	4	4	-1	4	-1
---	---	---	----	---	----

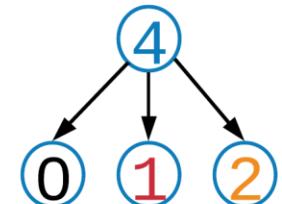
frontier

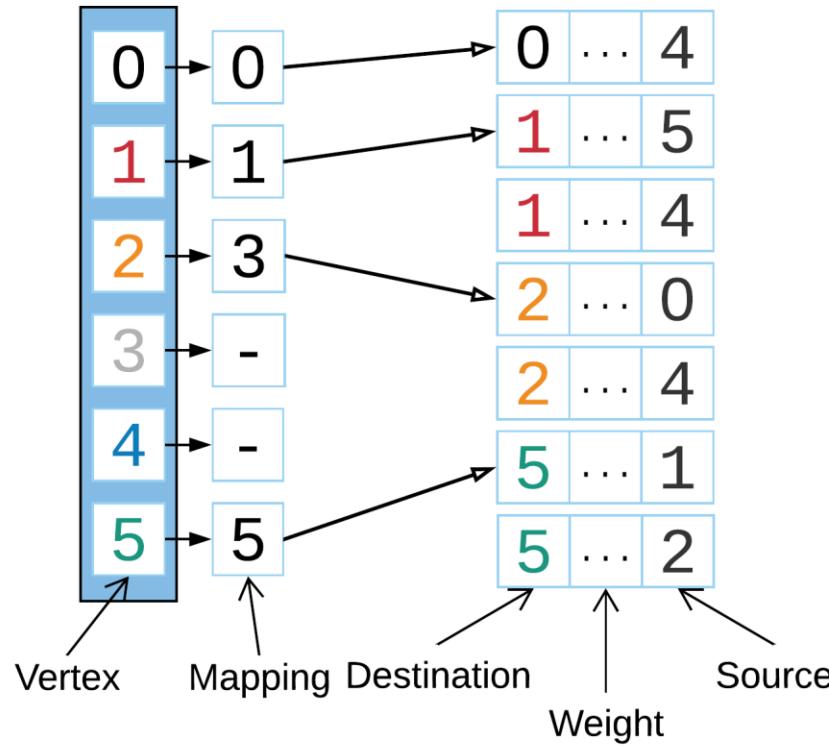
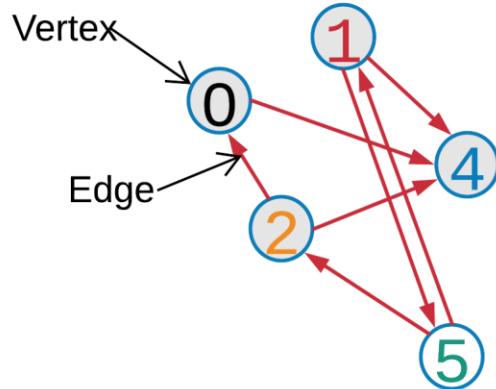
0	0	0	0	1	0
---	---	---	---	---	---

next

1	1	1	0	0	0
---	---	---	---	---	---

parent





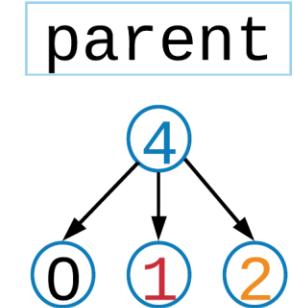
Source [4]

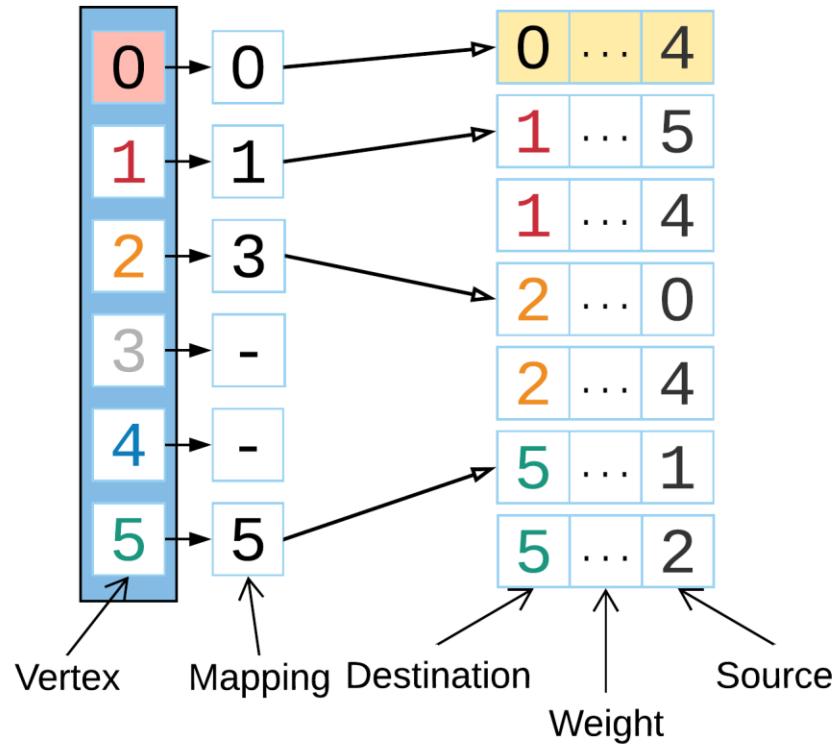
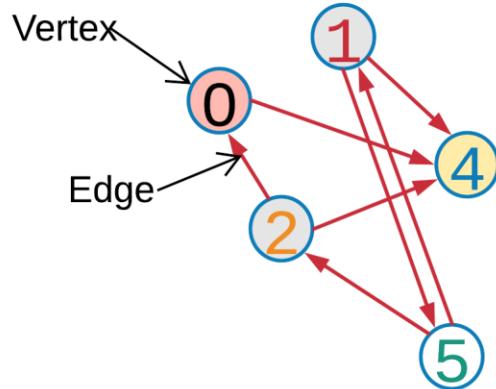
0 1 2 3 4 5

parent [4 4 4 -1 4 -1]

frontier [1 1 1 0 0 0]

next [0 0 0 0 0 0]





Source

4

0	1	2	3	4	5
---	---	---	---	---	---

parent

4	4	4	-1	4	-1
---	---	---	----	---	----

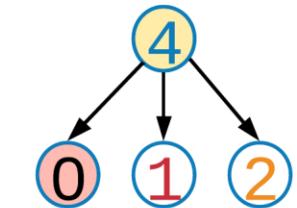
frontier

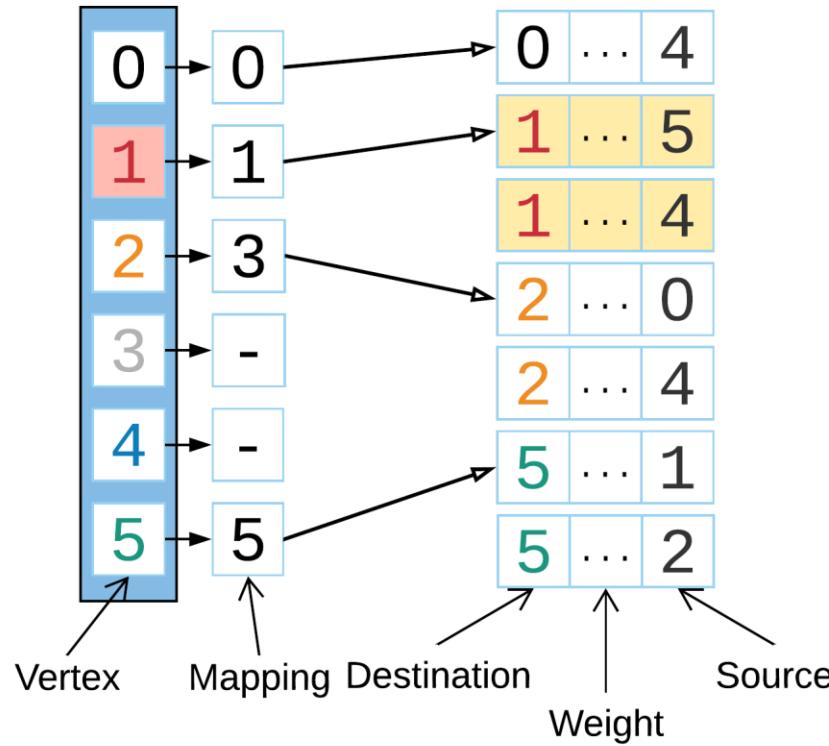
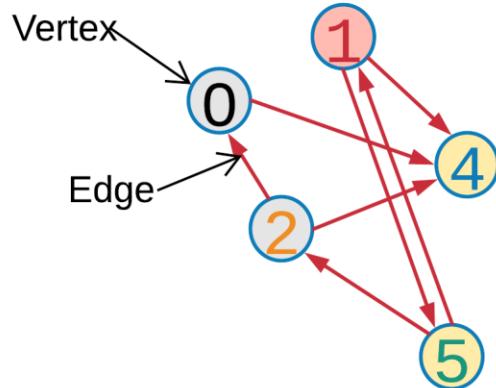
1	1	1	0	0	0
---	---	---	---	---	---

next

0	0	0	0	0	0
---	---	---	---	---	---

parent





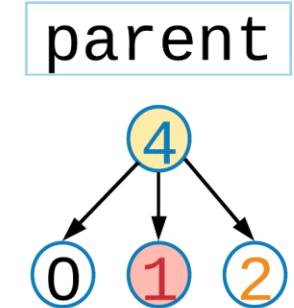
Source [4]

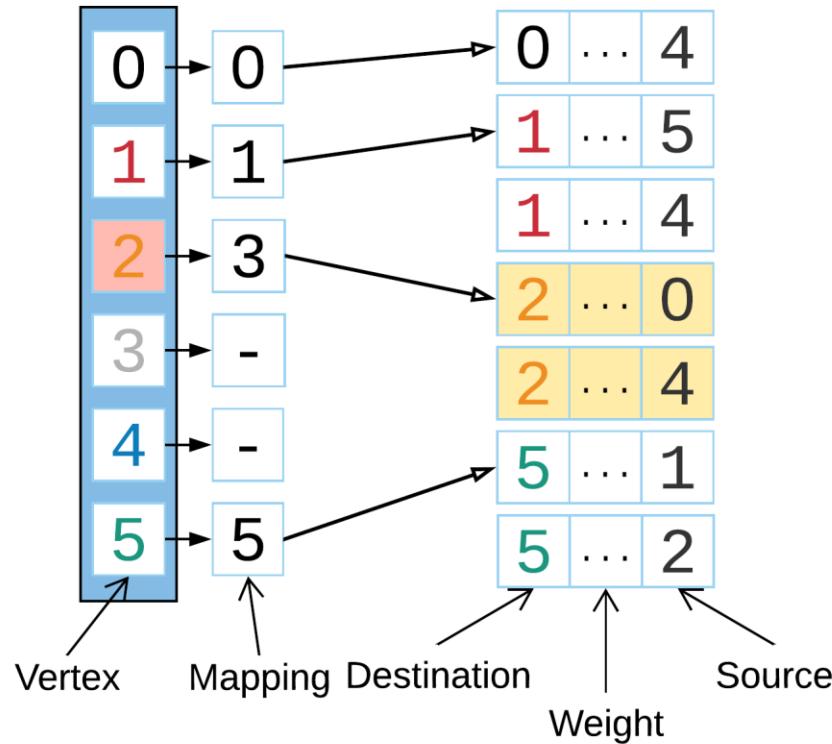
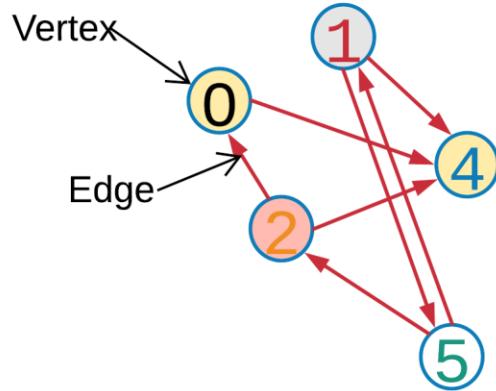
0 1 2 3 4 5

parent [4 4 4 -1 4 -1]

frontier [1 1 1 0 0 0]

next [0 0 0 0 0 0]





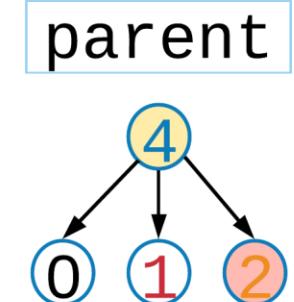
Source [4]

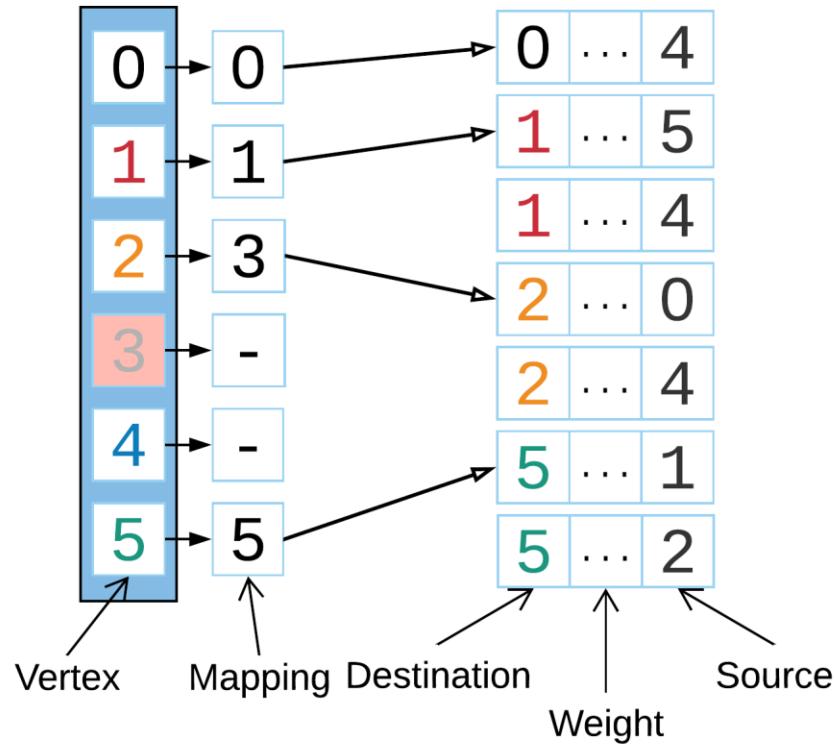
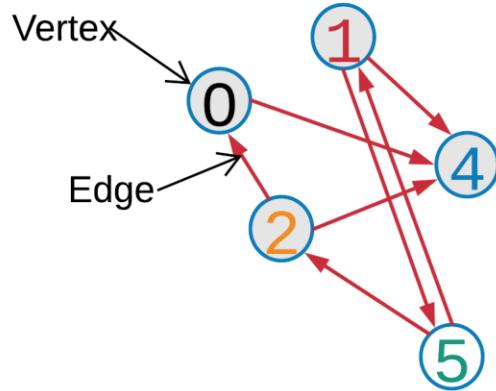
0 1 2 3 4 5

parent [4 4 4 -1 4 -1]

frontier [1 1 1 0 0 0]

next [0 0 0 0 0 0]





Source

4					
0	1	2	3	4	5

parent

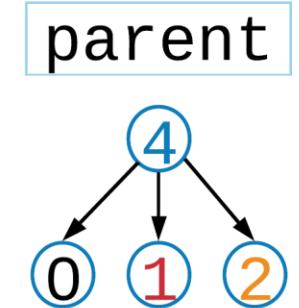
4	4	4	-1	4	-1
---	---	---	----	---	----

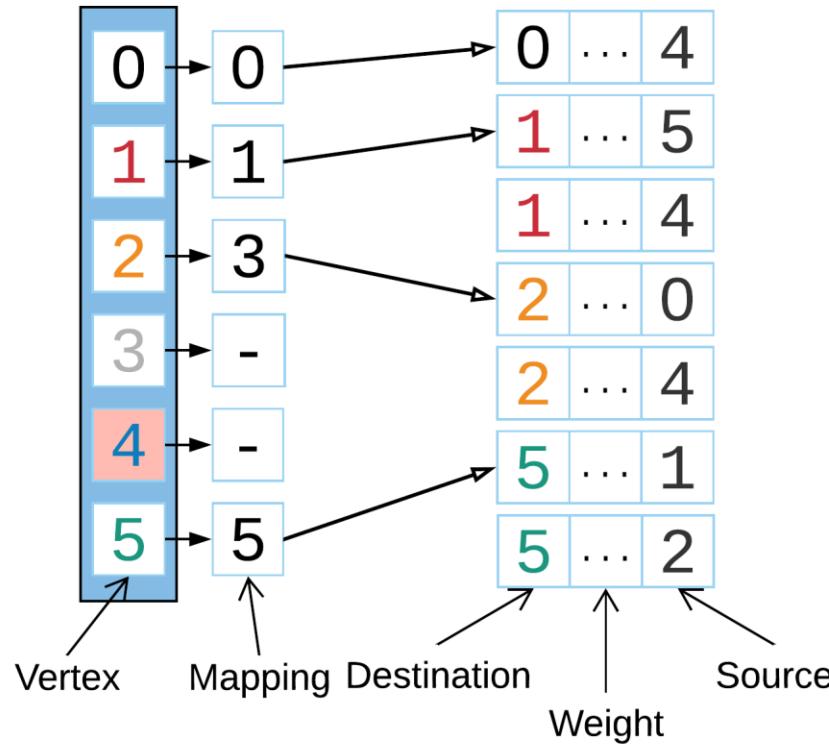
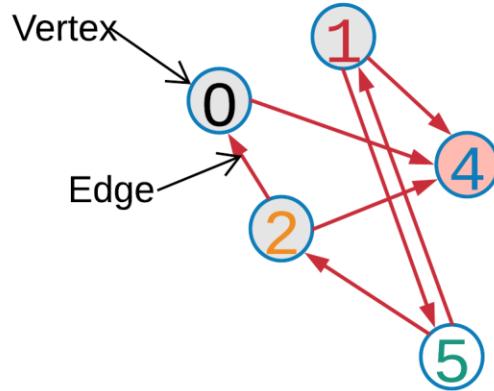
frontier

1	1	1	0	0	0
---	---	---	---	---	---

next

0	0	0	0	0	0
---	---	---	---	---	---





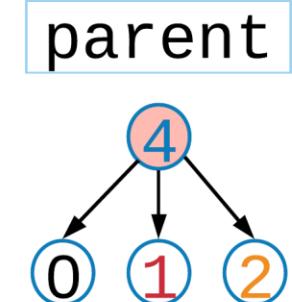
Source [4]

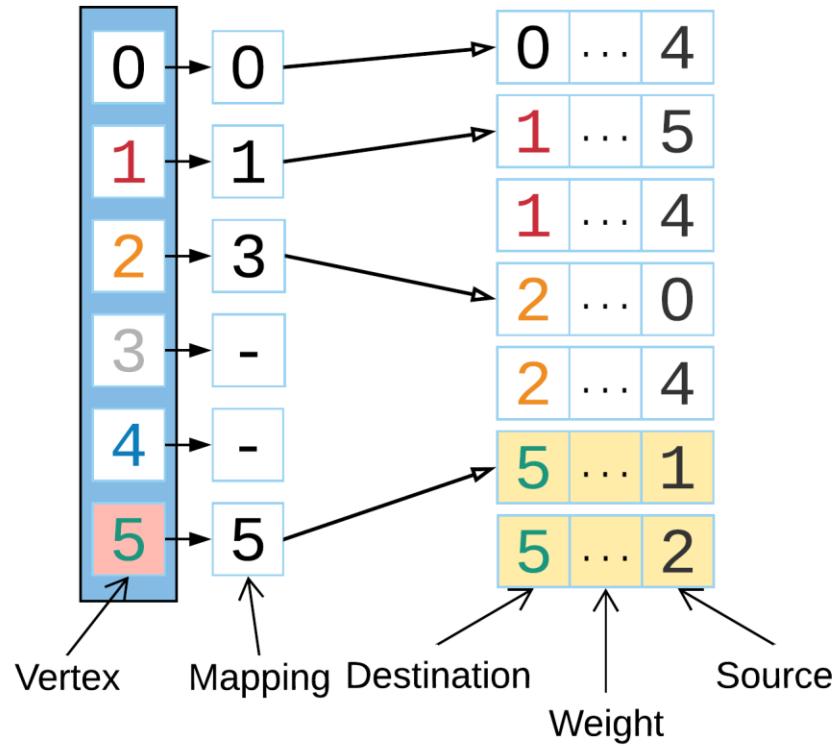
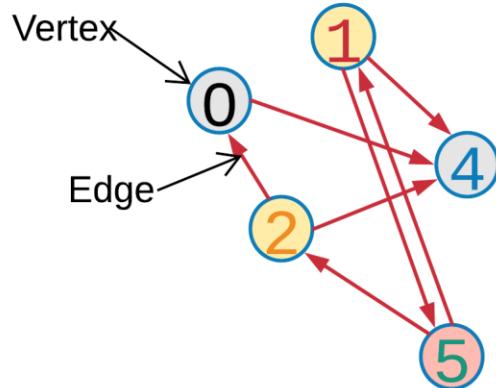
0 1 2 3 4 5

parent [4 4 4 -1 4 -1]

frontier [1 1 1 0 0 0]

next [0 0 0 0 0 0]





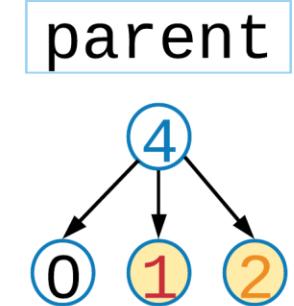
Source [4]

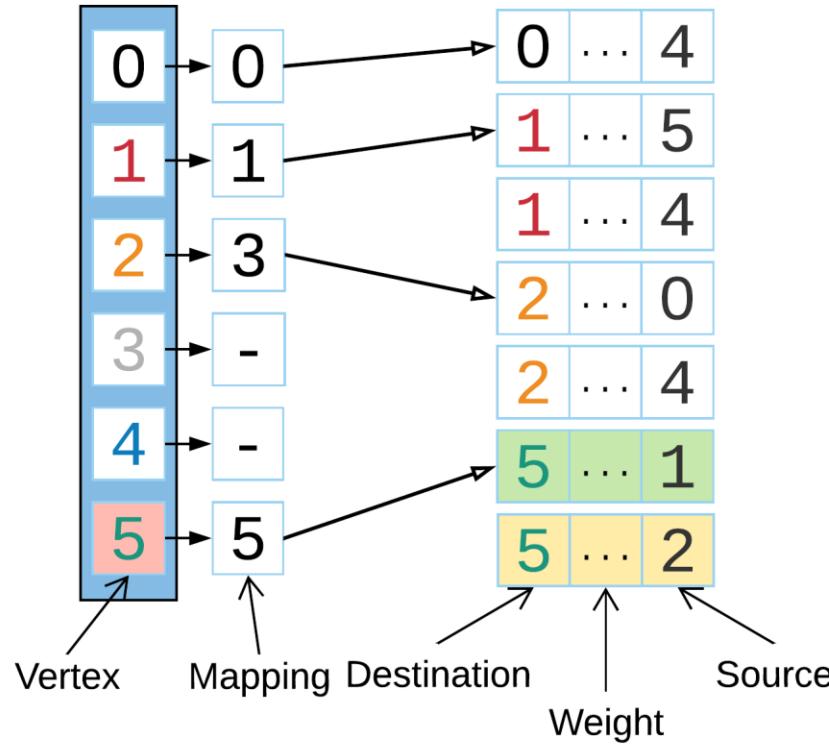
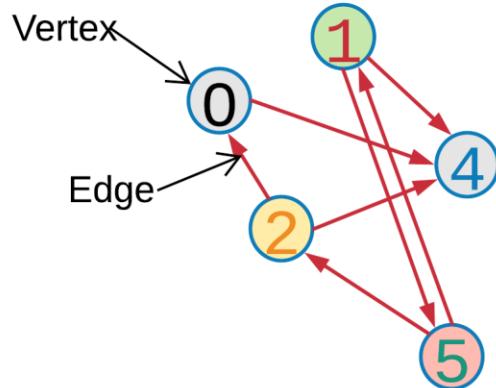
0 1 2 3 4 5

parent [4 4 4 -1 4 -1]

frontier [1 1 1 0 0 0]

next [0 0 0 0 0 0]





Source

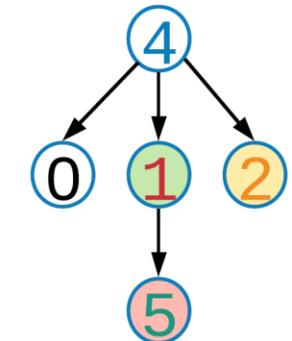
4

0 1 2 3 4 5

parent

4 4 4 -1 4 1

parent

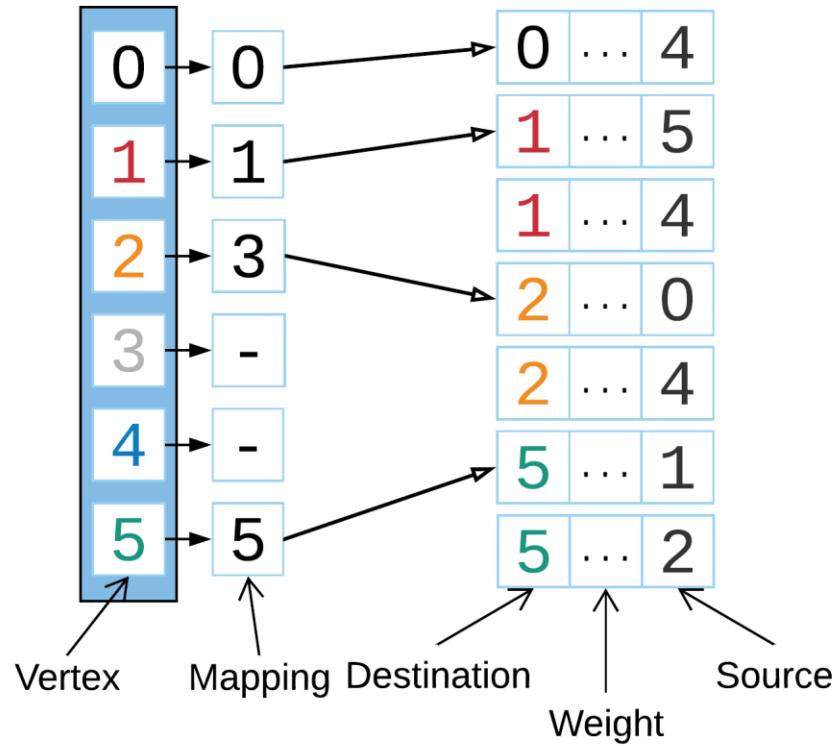
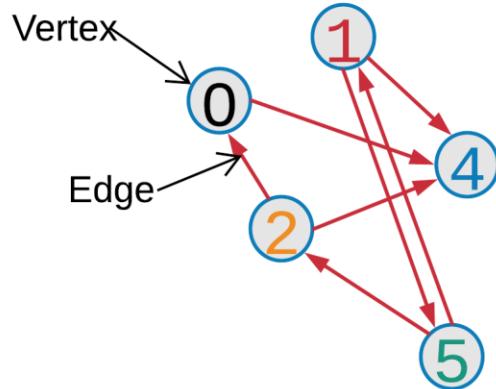


frontier

1 1 1 0 0 0

next

0 0 0 0 0 1



Source

4

0 1 2 3 4 5

parent

4 4 4 -1 4 1

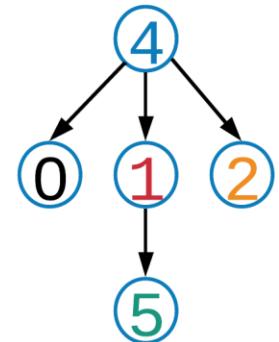
frontier

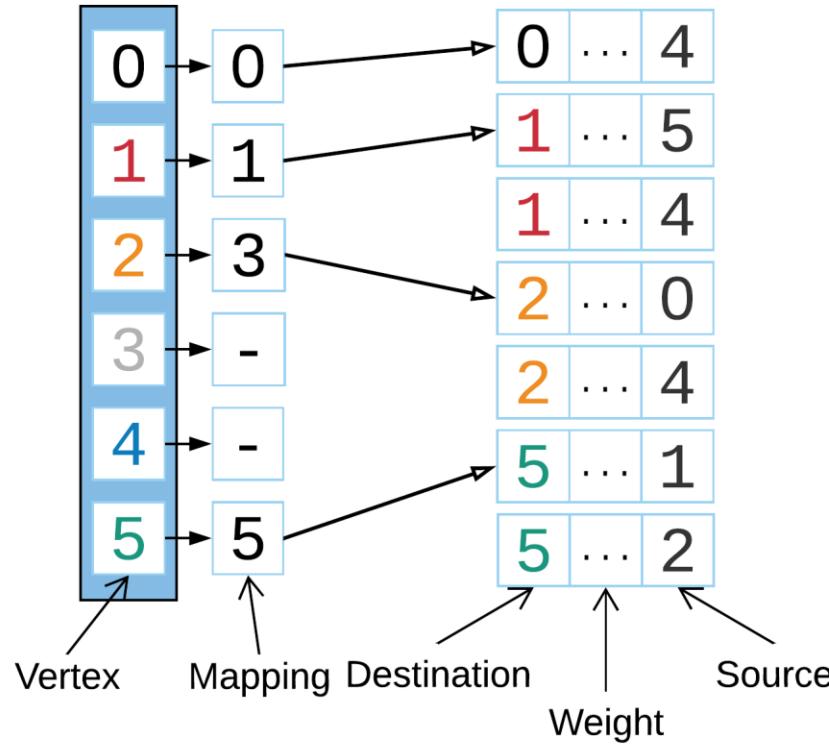
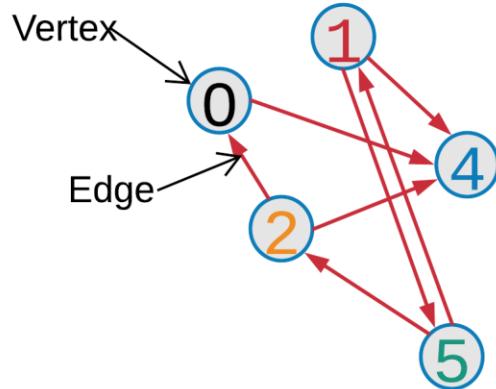
1 1 1 0 0 0

next

0 0 0 0 0 1

parent





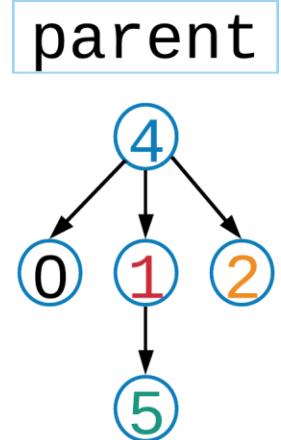
Source [4]

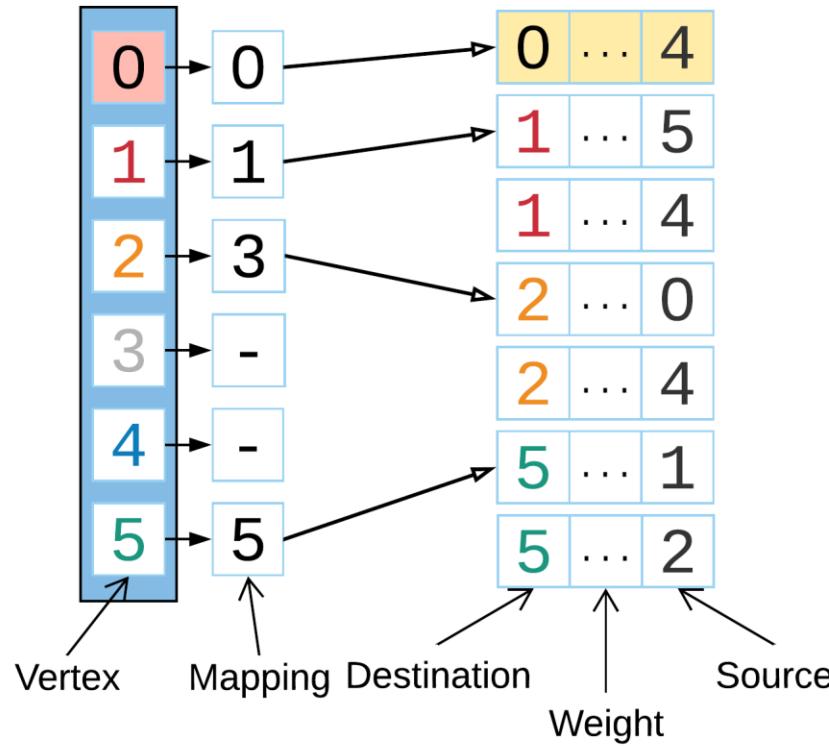
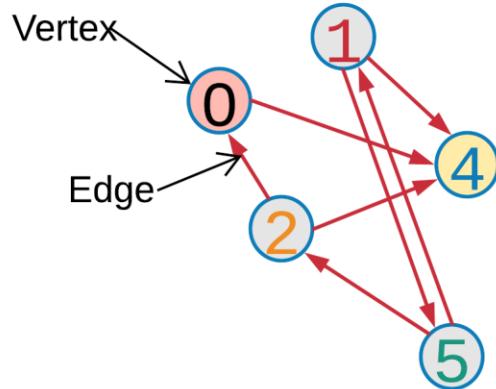
0 1 2 3 4 5

parent [4 4 4 -1 4 1]

frontier [0 0 0 0 0 1]

next [0 0 0 0 0 0]





Source

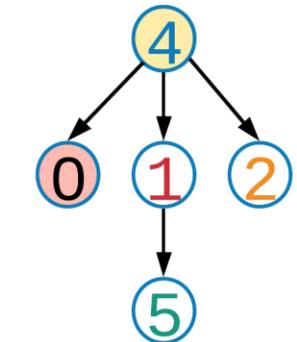
4

0	1	2	3	4	5
---	---	---	---	---	---

parent

4	4	4	-1	4	1
---	---	---	----	---	---

parent

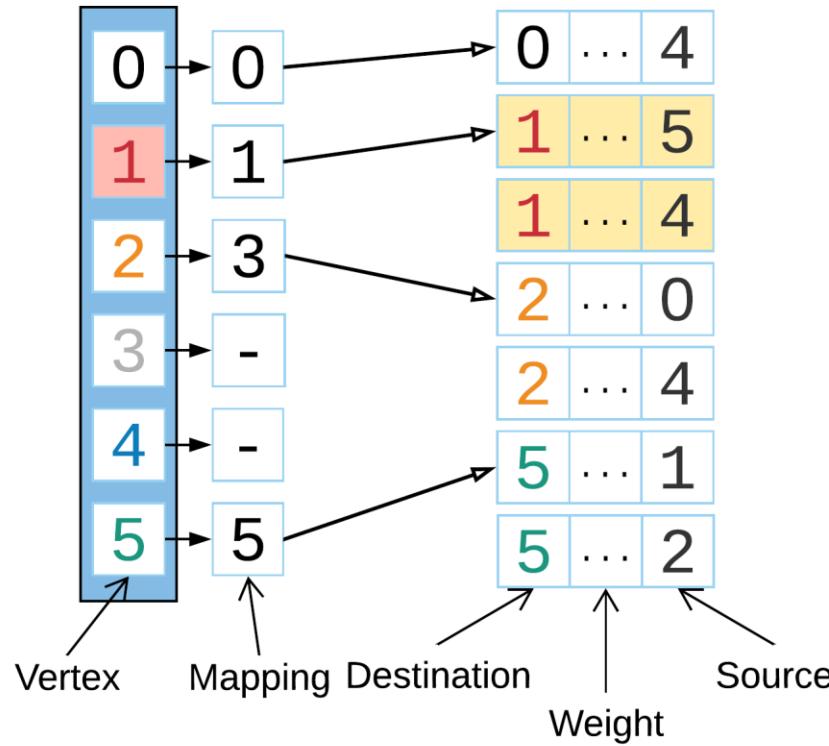
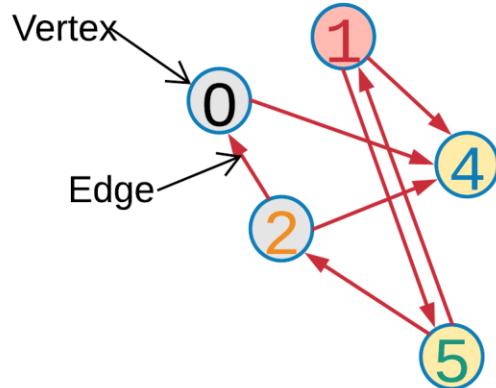


frontier

0	0	0	0	0	1
---	---	---	---	---	---

next

0	0	0	0	0	0
---	---	---	---	---	---



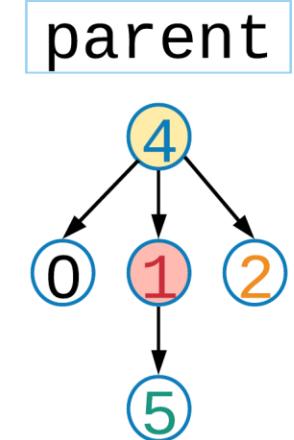
Source [4]

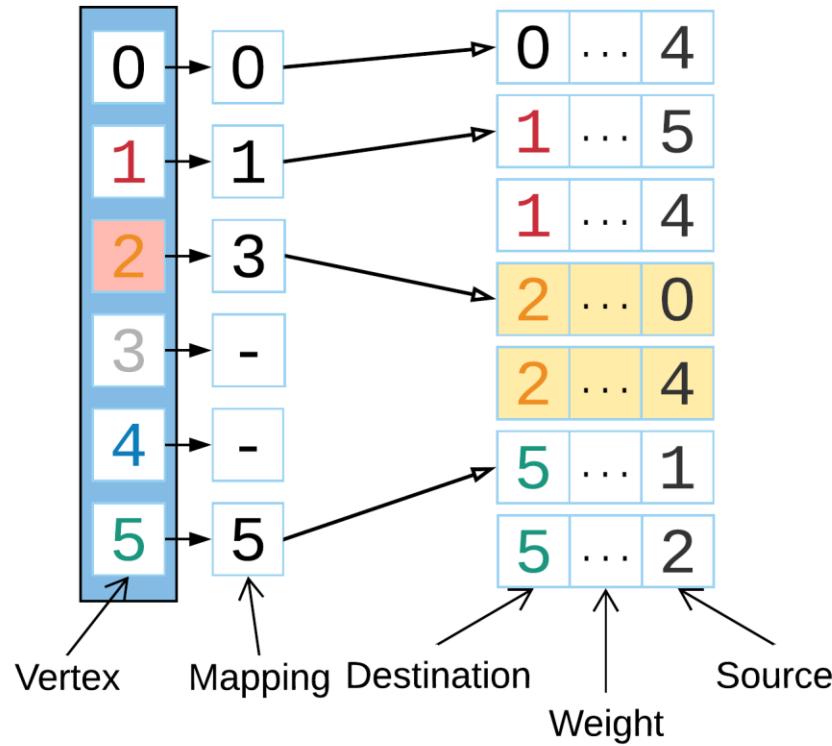
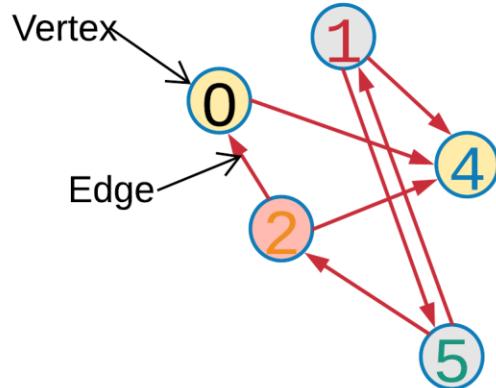
0 1 2 3 4 5

parent [4 4 4 -1 4 1]

0 0 0 0 0 1

next [0 0 0 0 0 0]





Source

4					
0	1	2	3	4	5

parent

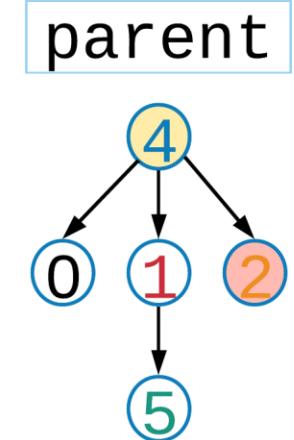
4	4	4	-1	4	1
---	---	---	----	---	---

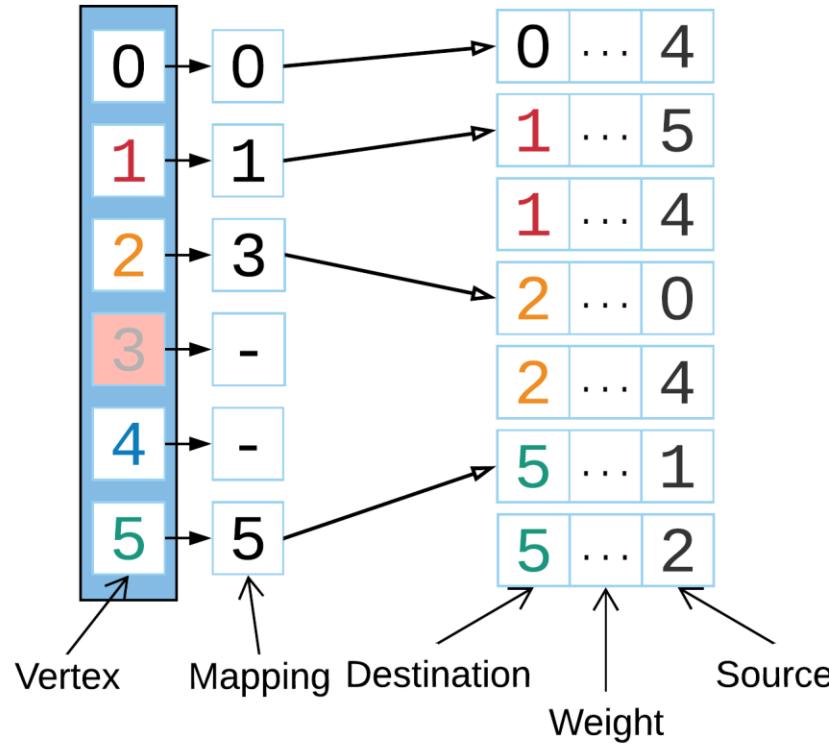
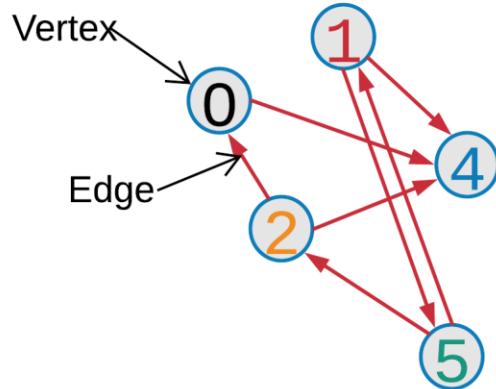
frontier

0	0	0	0	0	1
---	---	---	---	---	---

next

0	0	0	0	0	0
---	---	---	---	---	---





Source

4					
0	1	2	3	4	5

parent

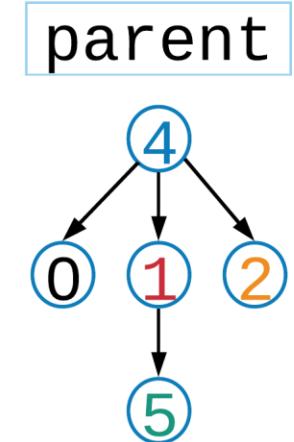
4	4	4	-1	4	1
---	---	---	----	---	---

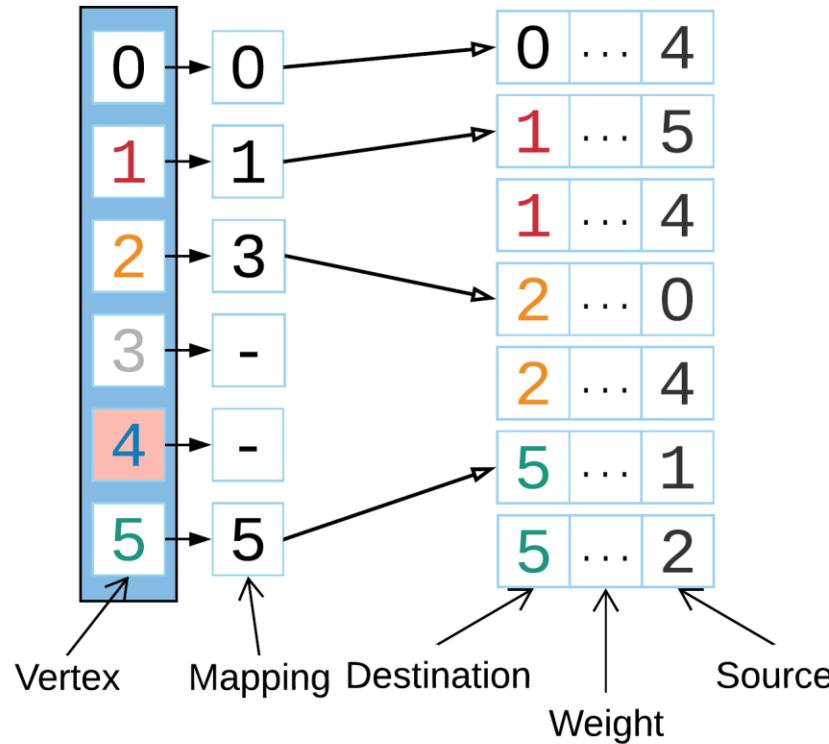
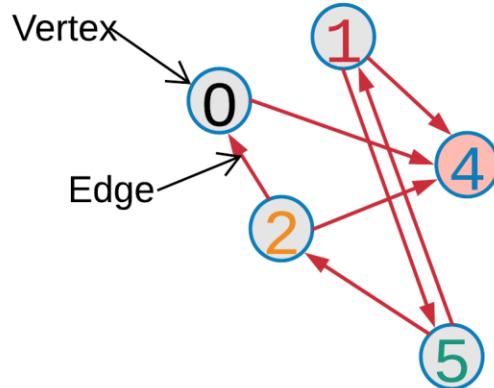
frontier

0	0	0	0	0	1
---	---	---	---	---	---

next

0	0	0	0	0	0
---	---	---	---	---	---





Source

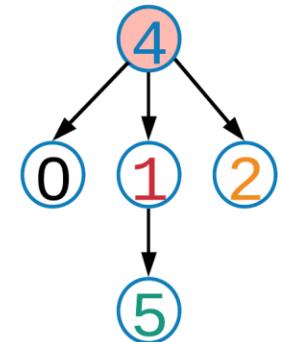
4

0	1	2	3	4	5
---	---	---	---	---	---

parent

4	4	4	-1	4	1
---	---	---	----	---	---

parent

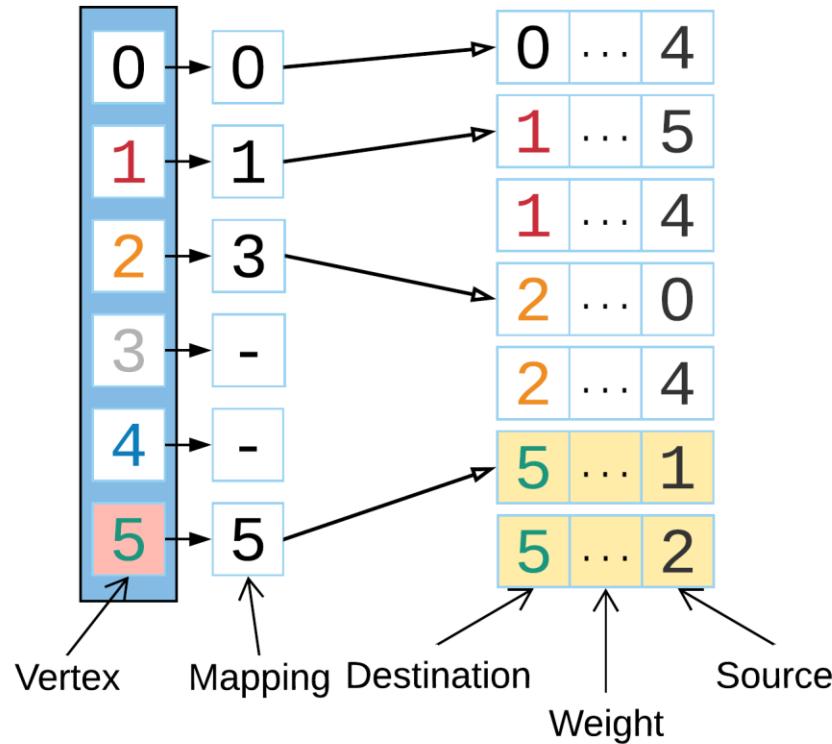
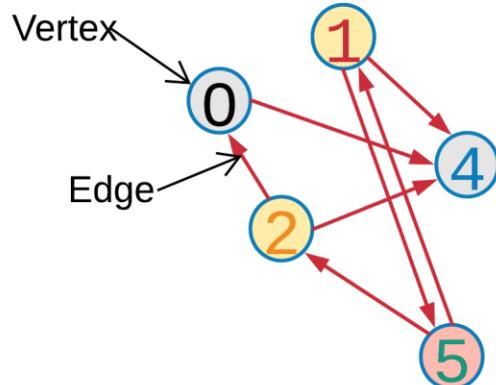


frontier

0	0	0	0	0	1
---	---	---	---	---	---

next

0	0	0	0	0	0
---	---	---	---	---	---



Source

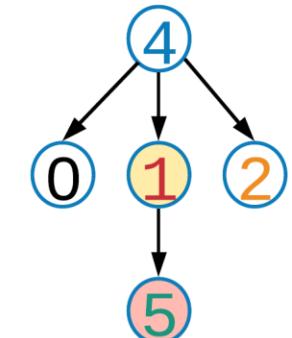
[4]

0 1 2 3 4 5

parent

[4] [4] [4] [-1] [4] [1]

parent

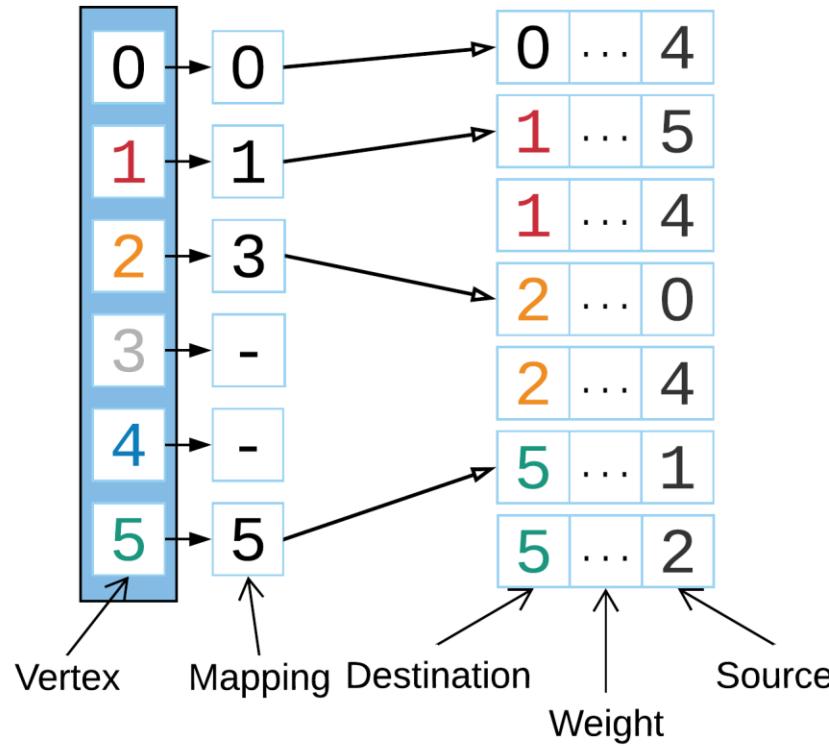
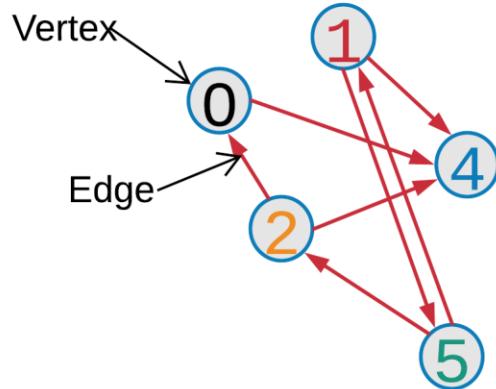


frontier

[0] [0] [0] [0] [0] [1]

next

[0] [0] [0] [0] [0] [0]



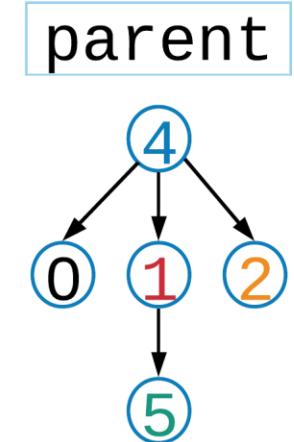
Source [4]

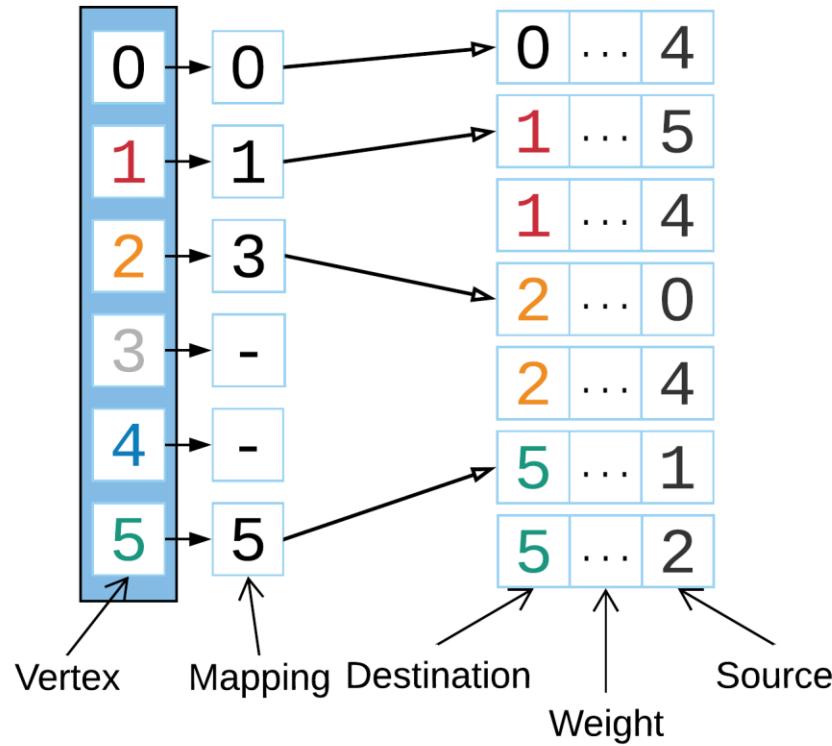
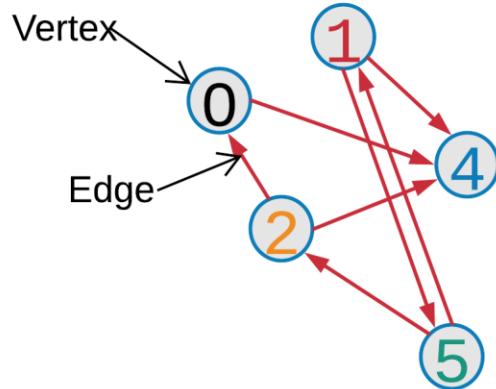
0 1 2 3 4 5

parent [4 4 4 -1 4 1]

frontier [0 0 0 0 0 1]

next [0 0 0 0 0 0]





Source

4					
0	1	2	3	4	5

parent

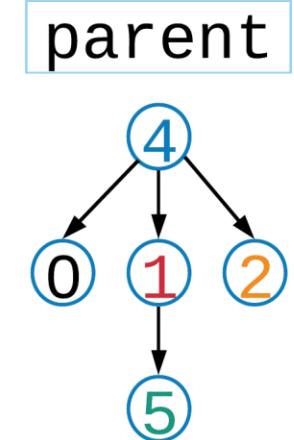
4	4	4	-1	4	1
---	---	---	----	---	---

frontier

0	0	0	0	0	0
---	---	---	---	---	---

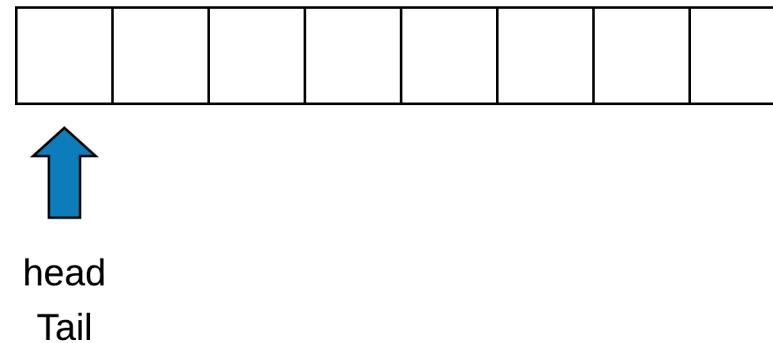
next

0	0	0	0	0	0
---	---	---	---	---	---



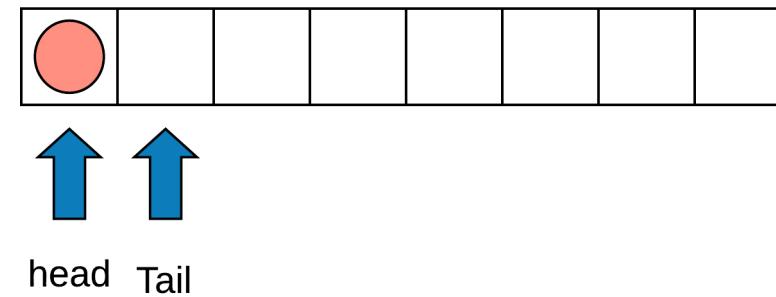
Array Queue (Empty)

Array-based FIFO queue of size $|V|$



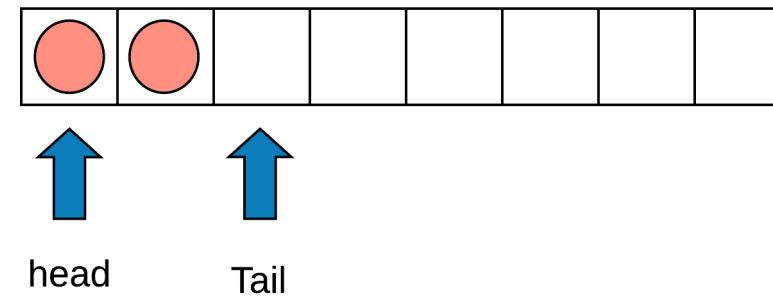
Array Queue (enqueue)

Array-based FIFO queue of size $|V|$



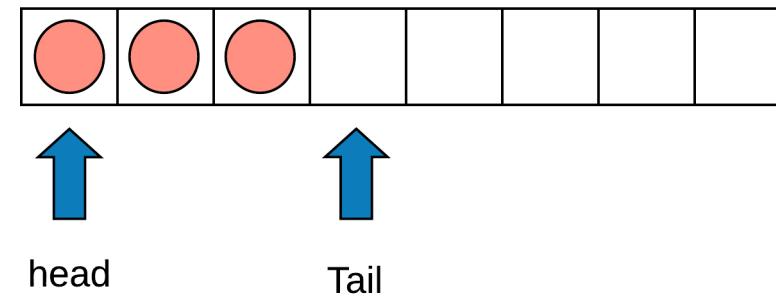
Array Queue (enqueue)

Array-based FIFO queue of size $|V|$



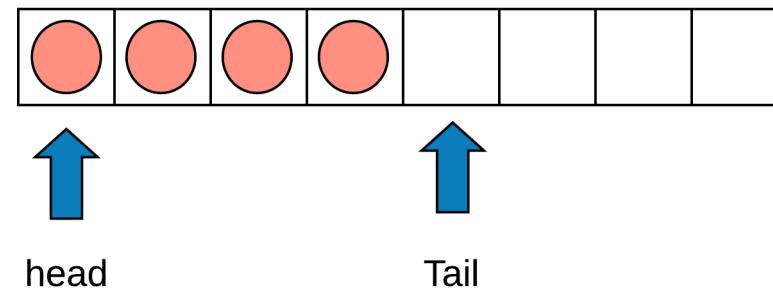
Array Queue (enqueue)

Array-based FIFO queue of size $|V|$



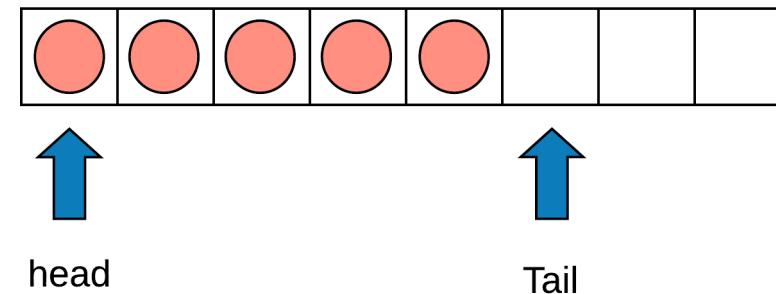
Array Queue (enqueue)

Array-based FIFO queue of size $|V|$



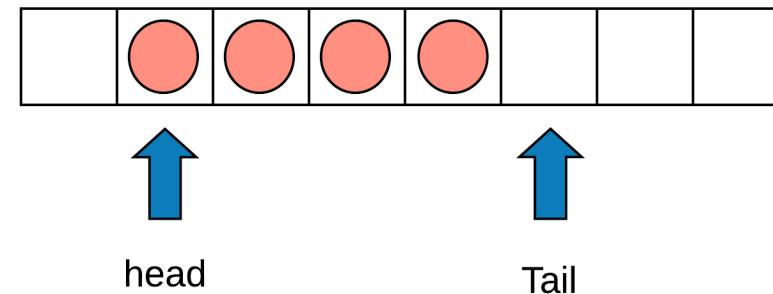
Array Queue (enqueue)

Array-based FIFO queue of size $|V|$



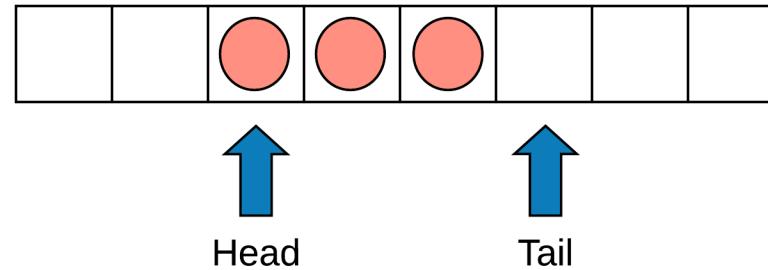
Array Queue (dequeue)

Array-based FIFO queue of size $|V|$



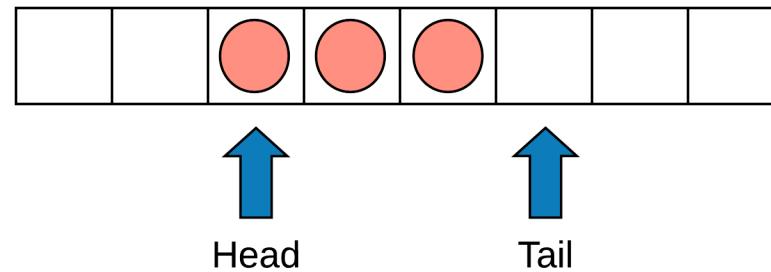
Array Queue (dequeue)

Array-based FIFO queue of size $|V|$

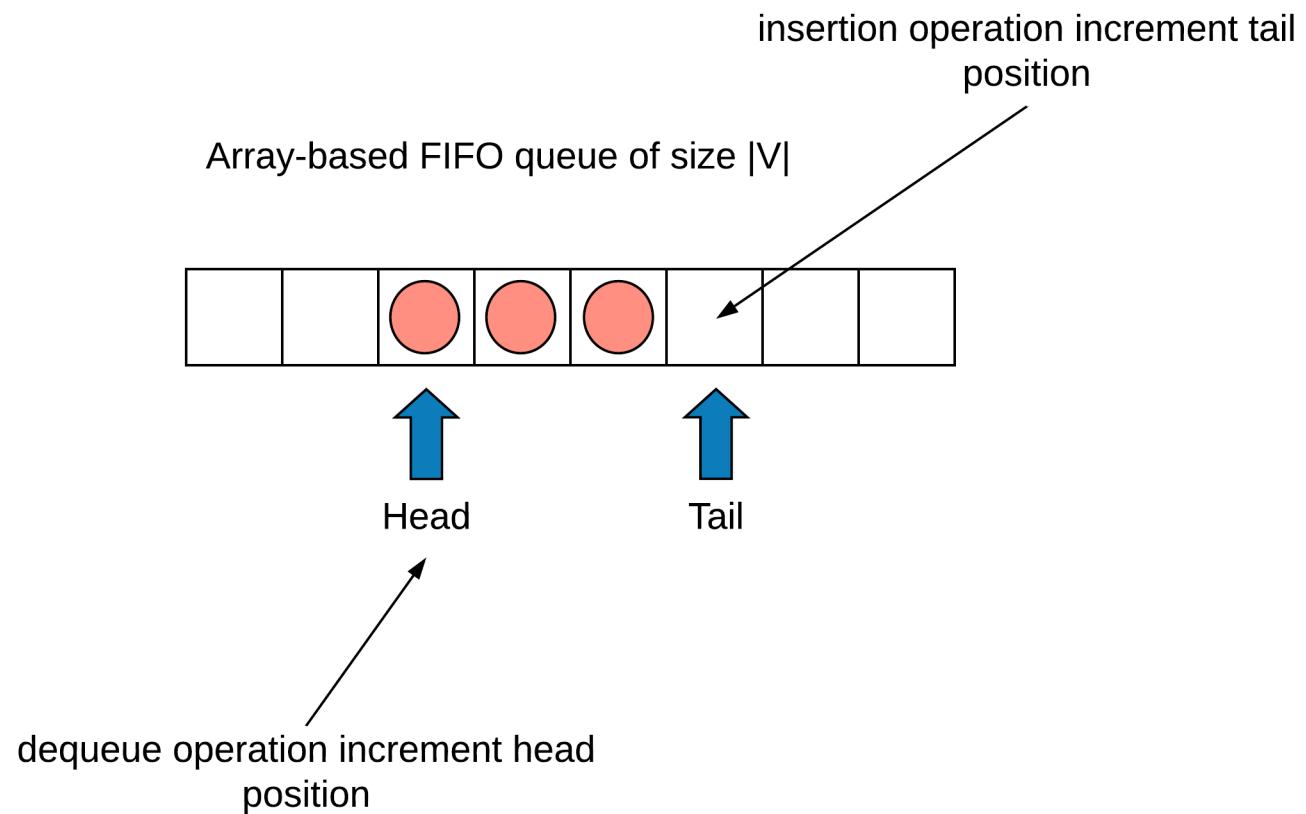


Array Queue

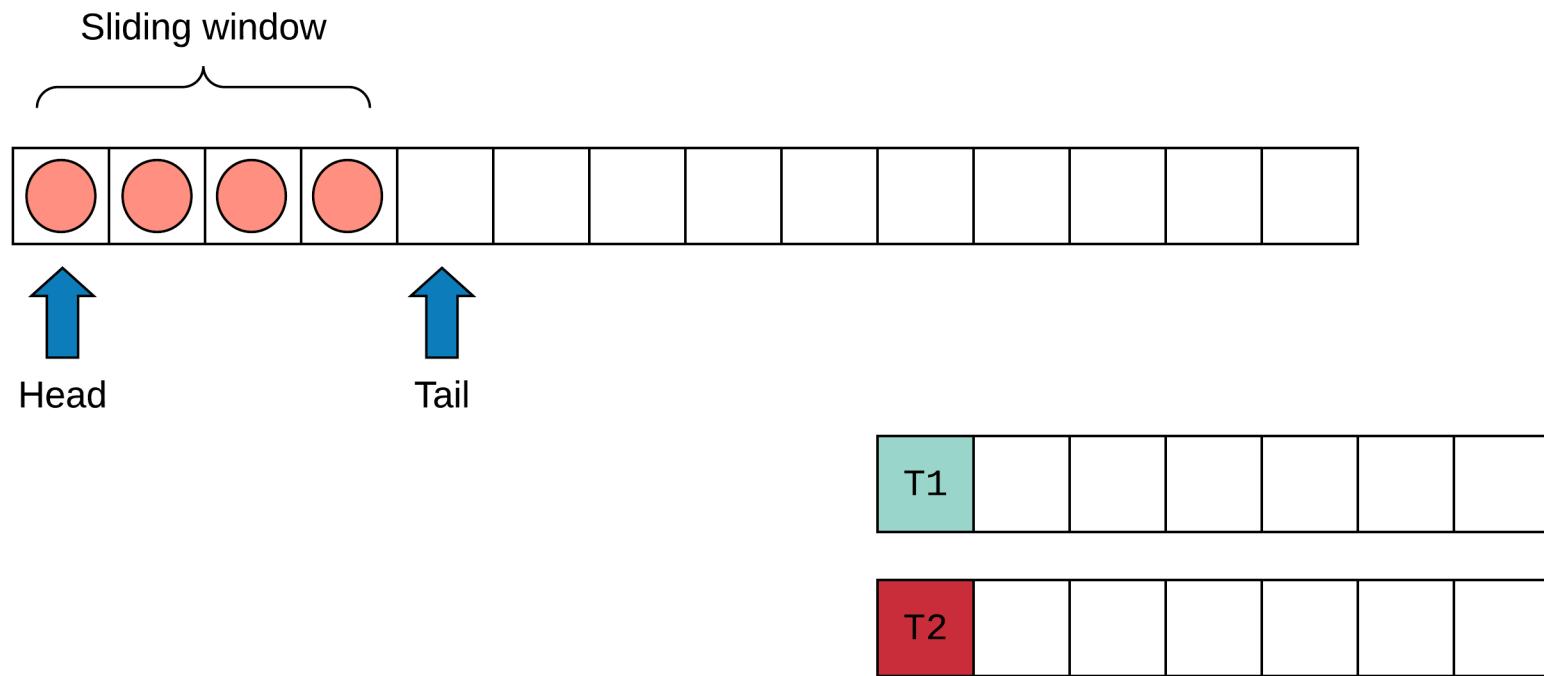
Array-based FIFO queue of size $|V|$



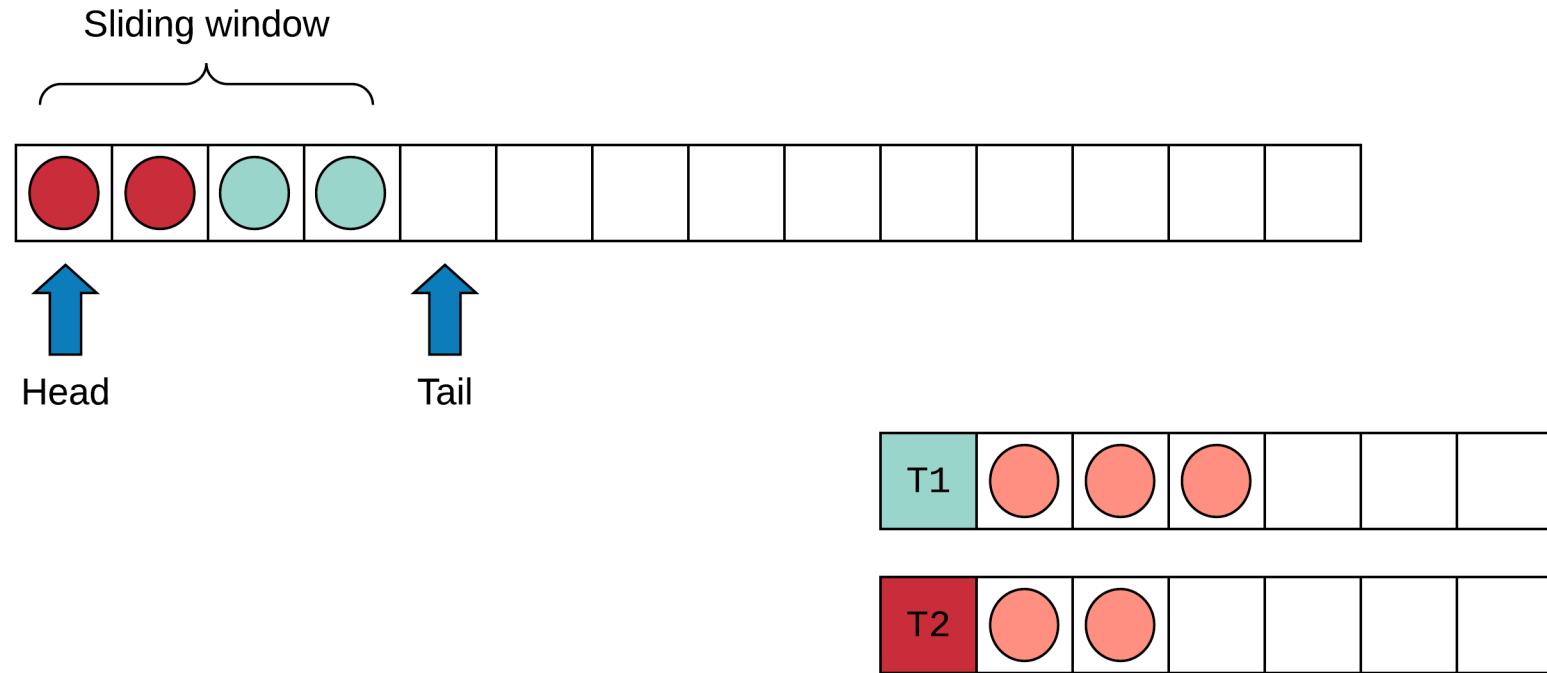
Array Queue



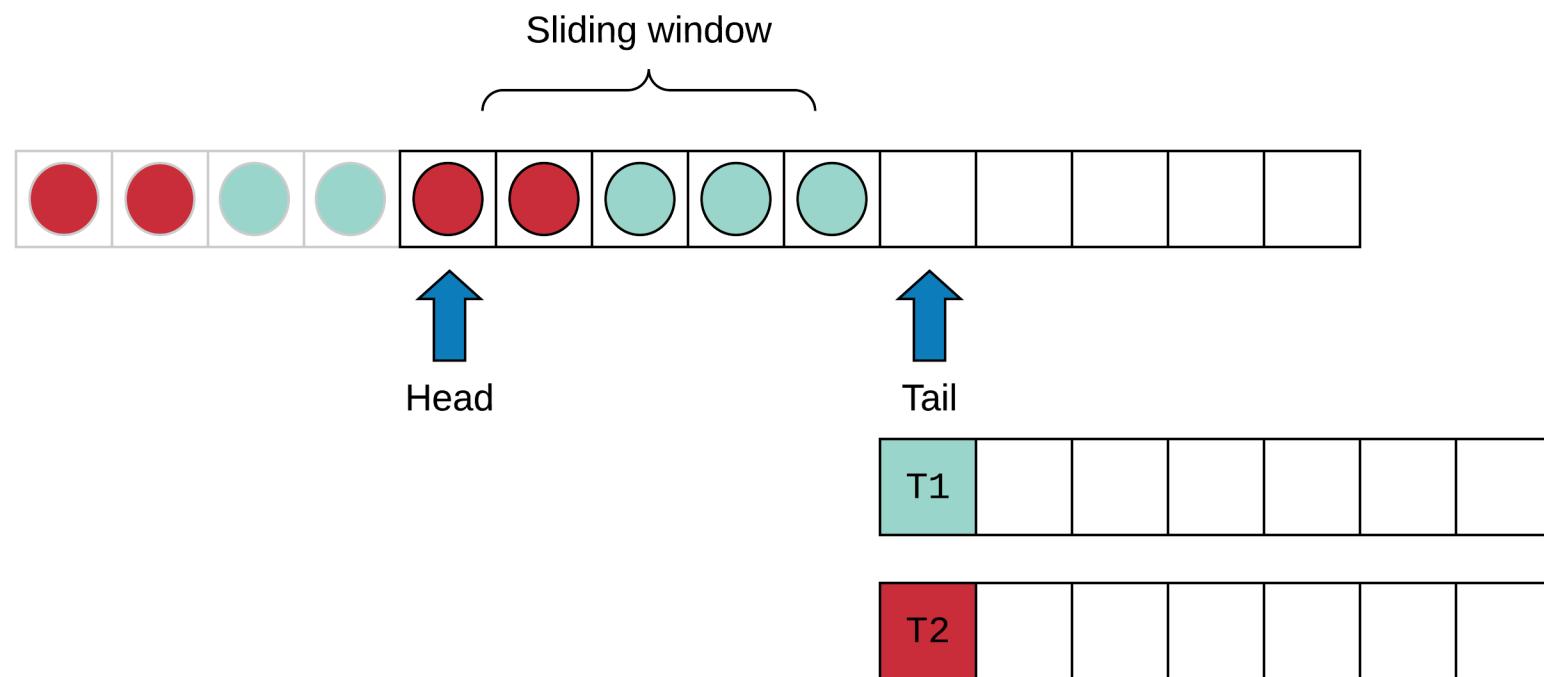
Parallel Sliding Queue



Parallel Sliding Queue



Array Queue



THANK YOU