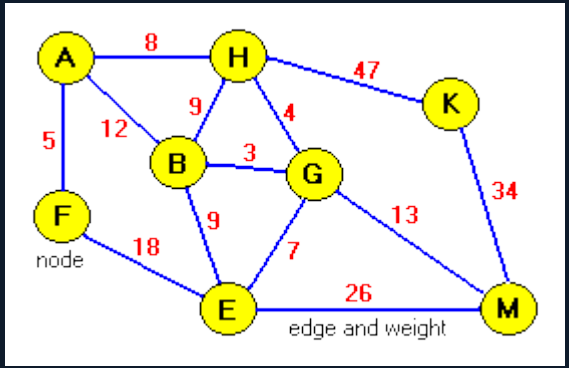
Graphs



Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Lecture Contents

- What is a graph?
- Connecting Data
- Nodes and Edges
- Different Graph Types
- Properties of Graphs
- Storing Graphs
- Inserting and removing from a graph

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

What is a Graph

- A graph is a data structure for storing the relationships between data points. In other words, how each piece of data is connected.

- There are two main parts of a graph
  - Vertices
    - These are the data points
  - Edges
    - These are the relationships / connections

Connecting data

Graph	Vertex	Edge
Communication	Telephones, Computers, Exchanges	Copper / Fiber-Optic cables
Transportation	Houses, Intersections, Airports, Docks	Roads, flight paths, shipping lanes
Facebook	People, Pages, Events	Friendships, Likes
World Wide Web	Web Pages	Hyperlinks
File System	Files	Folders, Shortcuts
Chess	Board Positions	Legal Moves

Connecting data

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Nodes and Edges

- The vertices of a graph can also be called Nodes
  - All a node has inside of it is some piece of data, dependant on the graph and a list of edges

- Edges contain the two Nodes they connect, along with information about the relationship between the data

Types of Graphs

- Defined by what kind of edges the graph has.

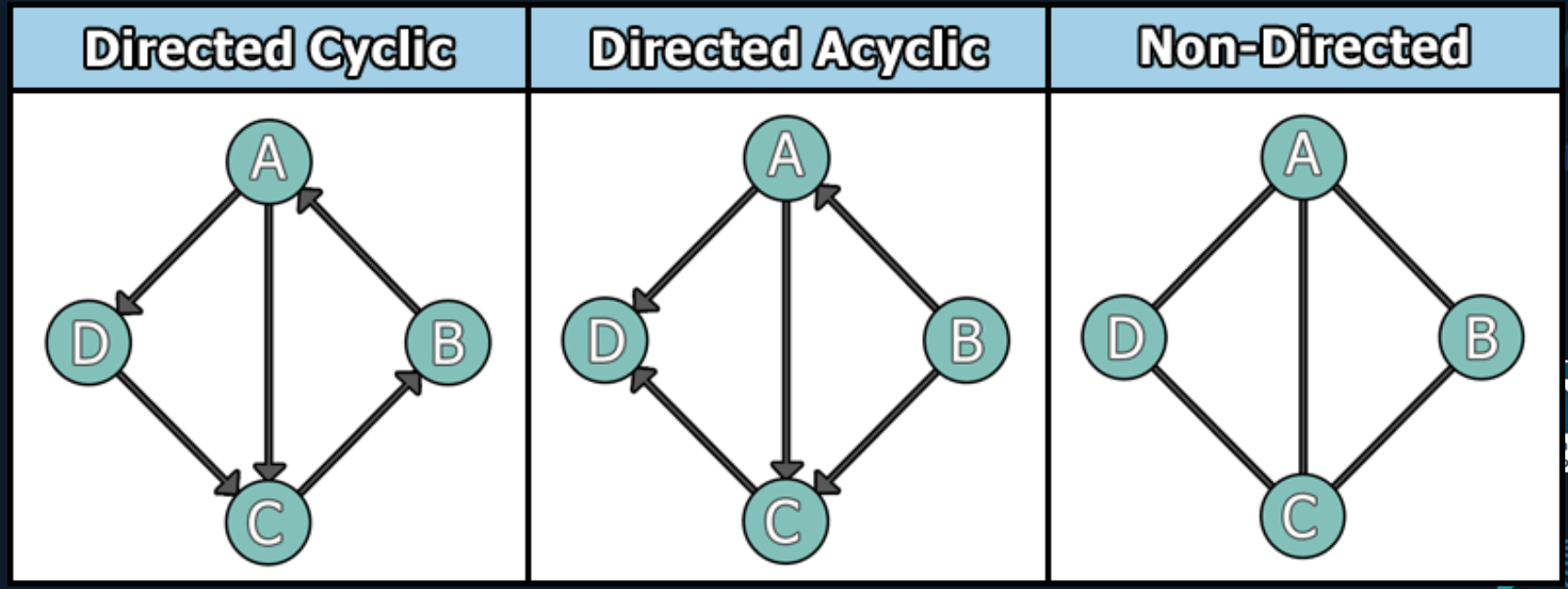
- Edges can be one way or two way.
  - Typically called directed or undirected.
  - For example, friends of Facebook are undirected, followers of twitter are directed.

Types of Graphs

- A graph that only has undirected edges is a non-directed graph.

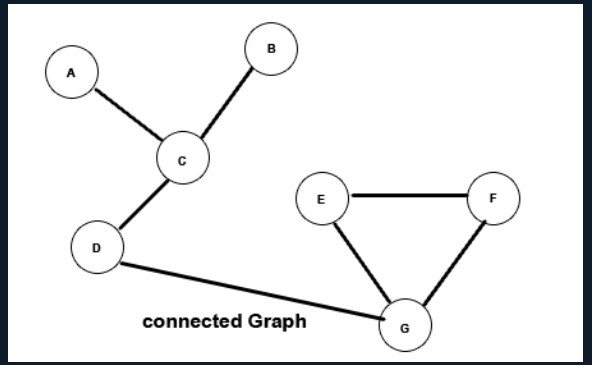
- If a graph has directed edges then it is a directed graph, or digraph.
  - If the digraph has a cycle, then it is a cyclic graph.
    - A cycle is a path of at least 3 nodes that starts and ends on the same node
  - If the digraph has no cycles it is an acyclic graph

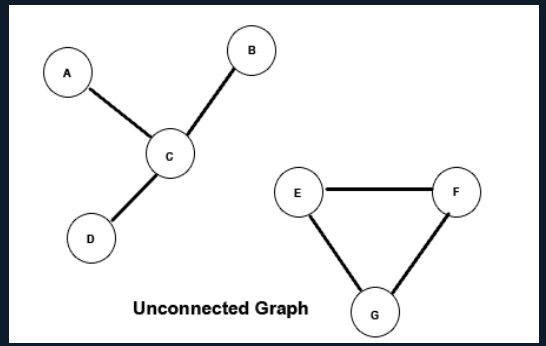
Types of Graphs



Other Properties of graphs

- A connected graph is a graph where, if you ignore directions, you can get from any node to any other node.
- The opposite of a connected graph is an unconnected or disjoint graph



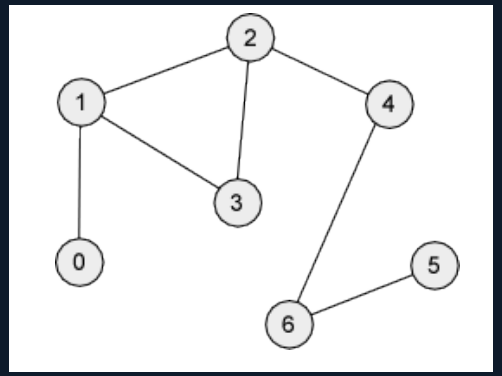


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

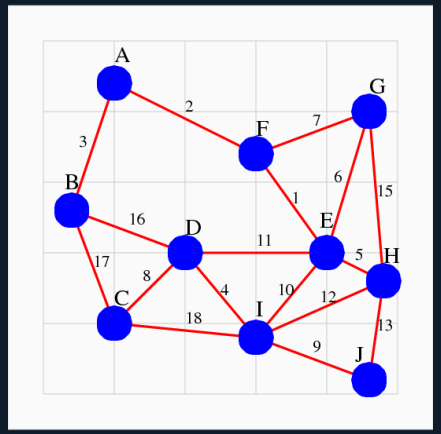
Other Properties of graphs

- The degree of a vertex is how many connections it has.
  - Node 5 has a degree of 1
  - Node 1 has a degree of 3



Other Properties of graphs

- A weighted graph is one where each edge has a cost associated with it.
- The connections on a social network are typically non-weighted.
- However, the links for graph of a map (where the nodes are places, and the edges are roads) might have each edge weight be the length of the road.



Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

How to Store Graphs

- There are two main ways to store graphs in a computer.
  - An Adjacency matrix
  - An Adjacency list

Adjacency Matrix

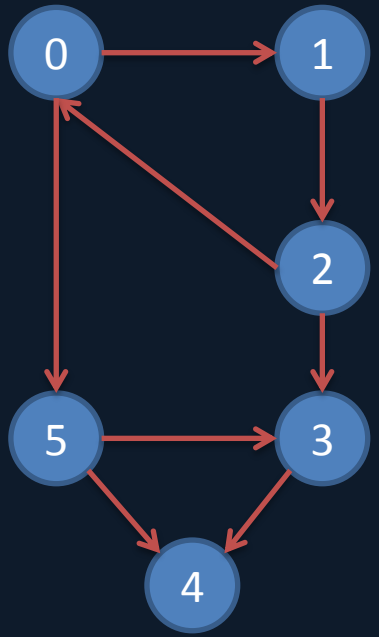
- You use an array to store the nodes and a matrix to store the edges.
- The array is just a list of each node
- The matrix is a square 2D array of connections between different nodes. The size is equal to the number of nodes in the graph.

Adjacency Matrix

- In the matrix, the row represents the starting node, and the column represents the ending node.

- The data stored at that index of the matrix contains information about the link.
  - The minimum information would be if the link exists or not, but could also include the cost or any other information the graph needs.

Adjacency Matrix



Array:

0	1	2	3	4	5

Matrix:

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

Adjacency Matrix

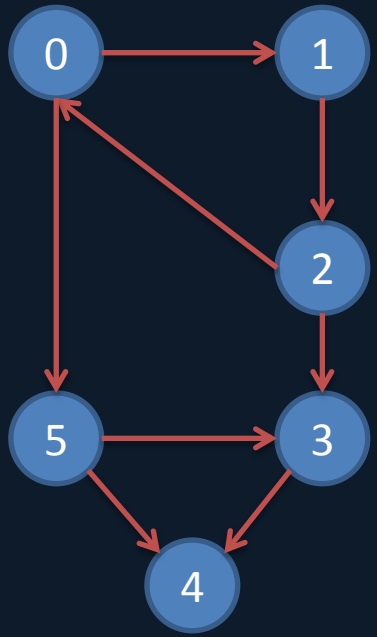
- Pros:
  - Really fast to check if two nodes are connected
  - Cache friendly
  - Some algorithms are easier to implement
- Cons
  - Costly to add or remove nodes
  - Can’t have more than one link between nodes
  - Wastes memory as you still need slots for edges that don’t exist

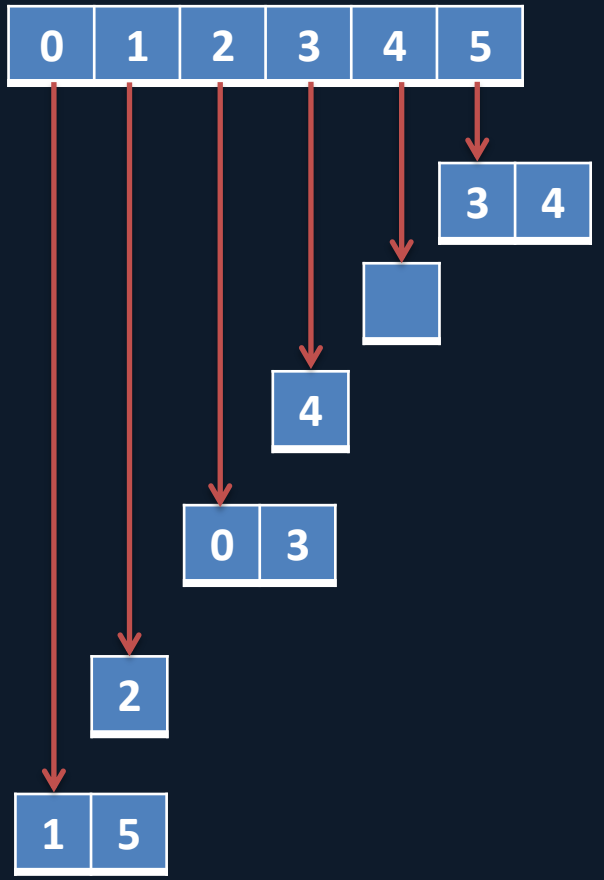
Adjacency List

- The other way to store a graph is an adjacency list.

- Each node simply contains a list of edges to the nodes they are connected to.

Adjacency List





Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Adjacency List

- Pros:
  - More Flexible
  - Uses less memory
- Cons:
  - Slower to find edges

Graph Operations

- We will now describe methods for some common operations you might want to perform on a graph.

- For the rest of these slides it is assumed we are using a directed graph with adjacency lists.

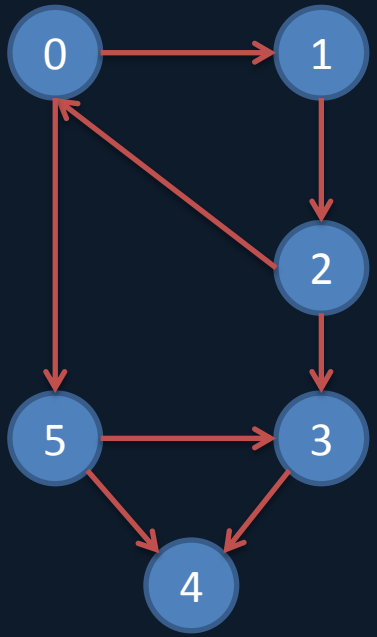
Graph Operations – Creating a Graph

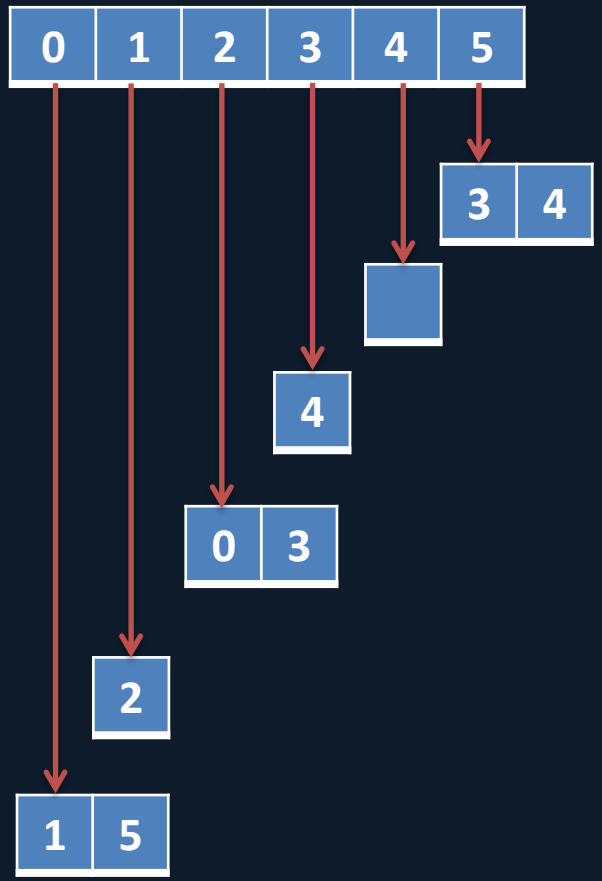
- Our Graph object simply contains an array of the nodes within it.
- Our Node Object contains our node data and an array of edges.
- Our Edge Object contains two Node pointers – its start and end nodes, and an integer for the cost of traversing the edge.

Graph Operations – Inserting a Node

- Inserting a node into our graph is simple.
- We can define that nodes inserted in our graph will begin disjoint to the rest of the graph.
- This means, all we need to do is create a new node and insert it into the node list in our graph object

Graph Operations – Inserting a Node

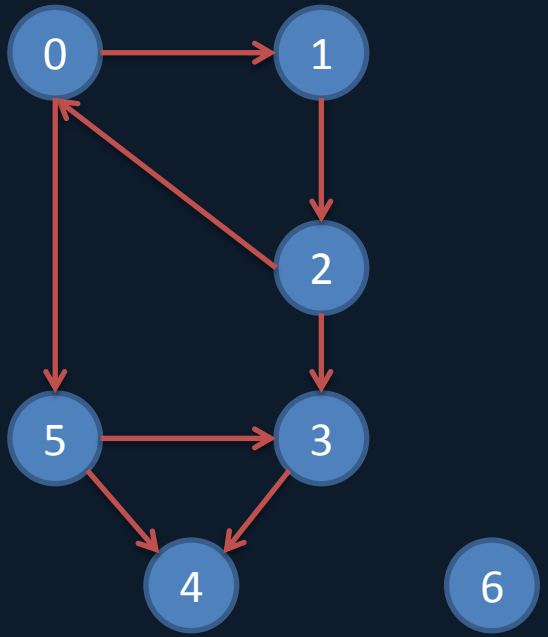


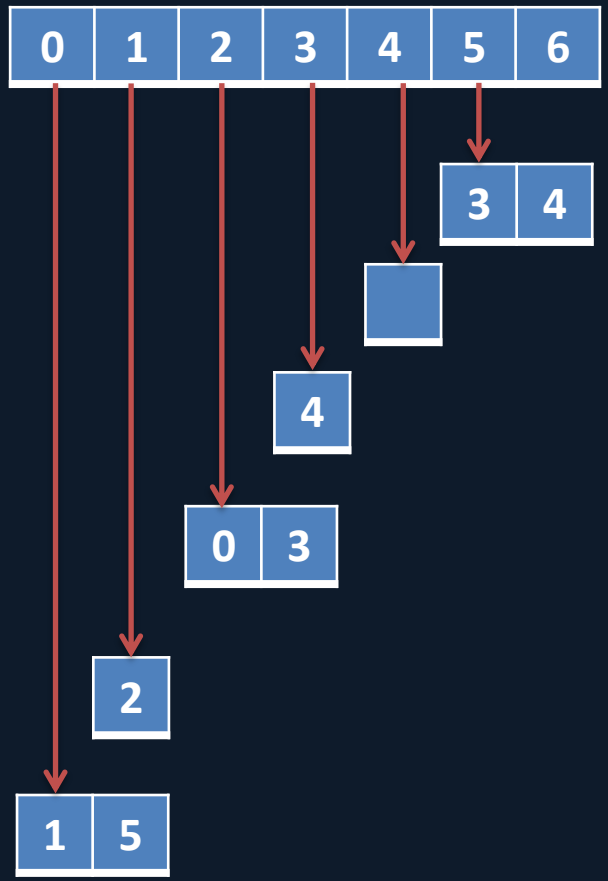


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Inserting a Node





Canberra Institute of Technology

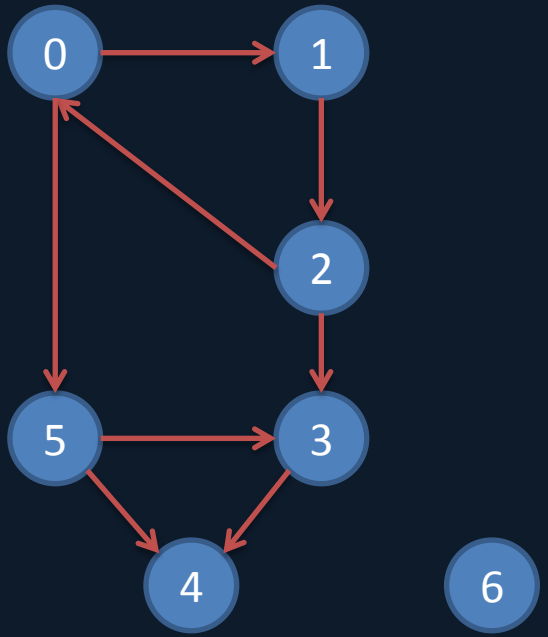
aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

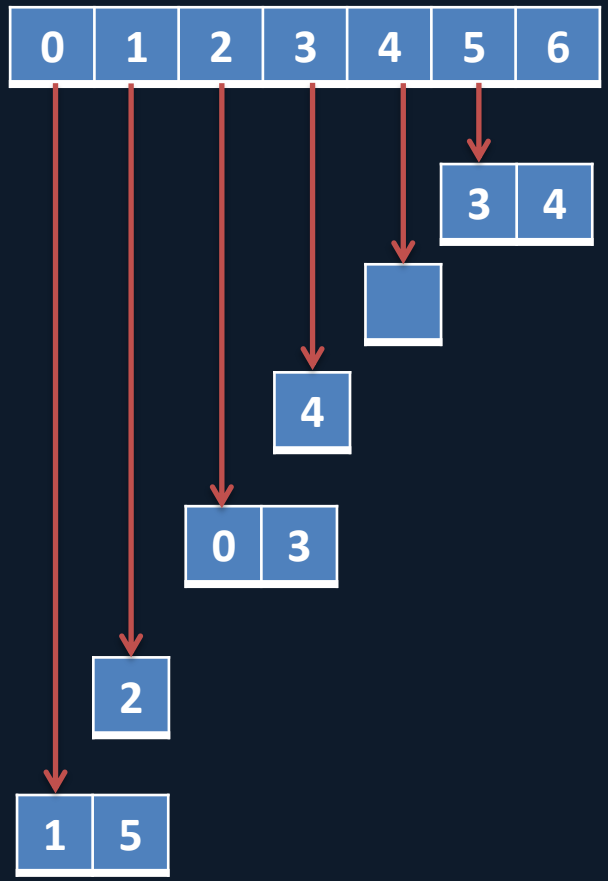
Graph Operations – Inserting an Edge

- Given two nodes, N1 and N2 inserting an edge from N1 to N2 also quite simple.

- We first create a new edge
- Then, we set its start and end pointers to point at N1 and N2 respectively.
- We then add the edge to N1's edge list.

Graph Operations – Inserting an Edge

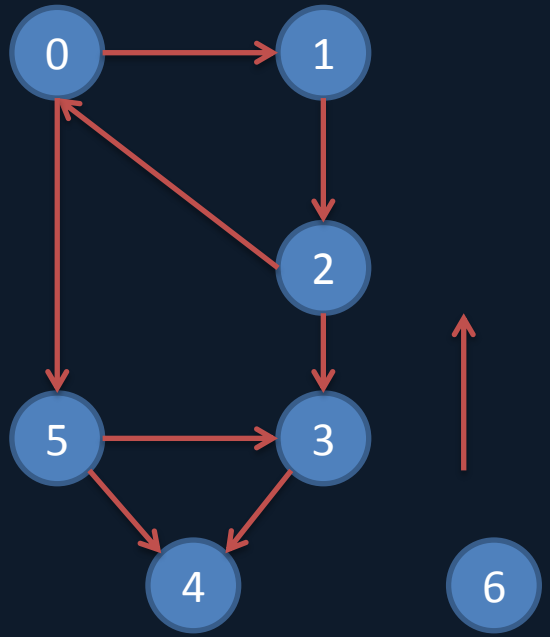


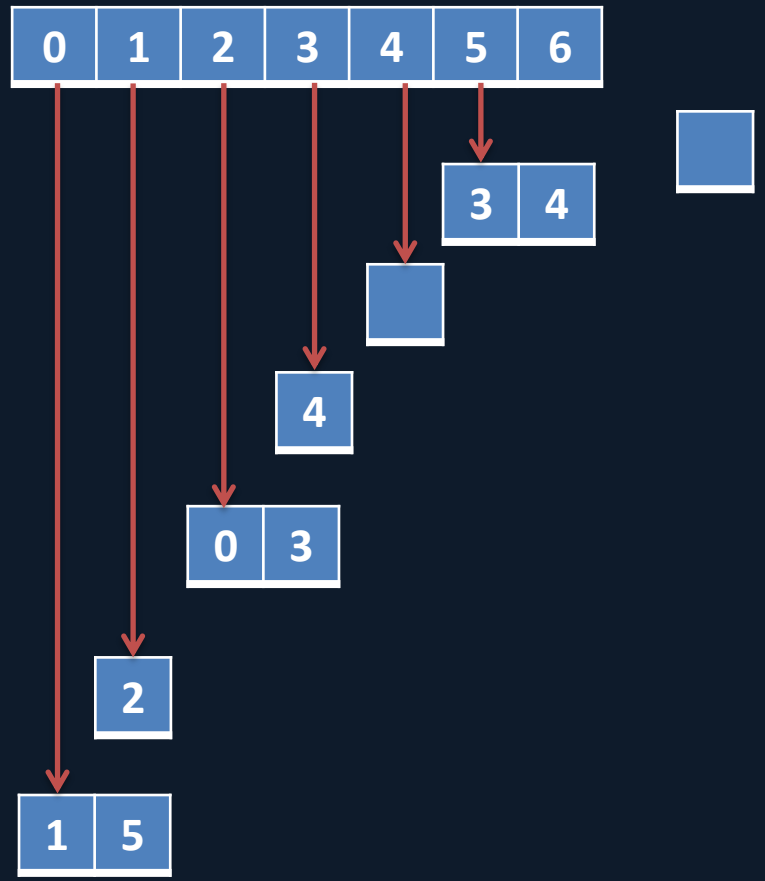


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Inserting an Edge

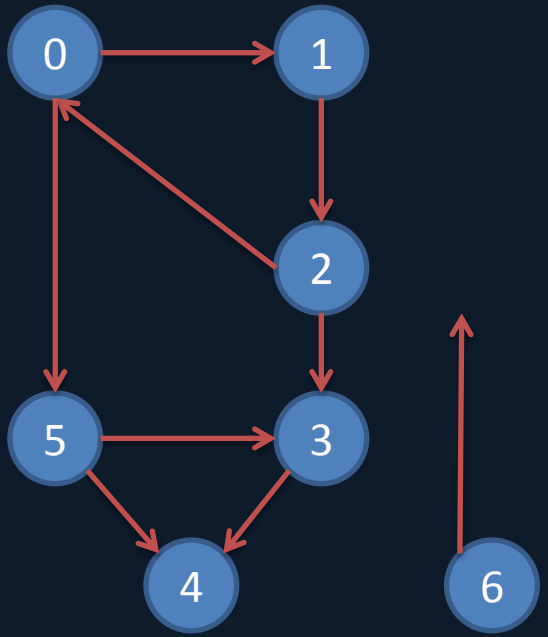


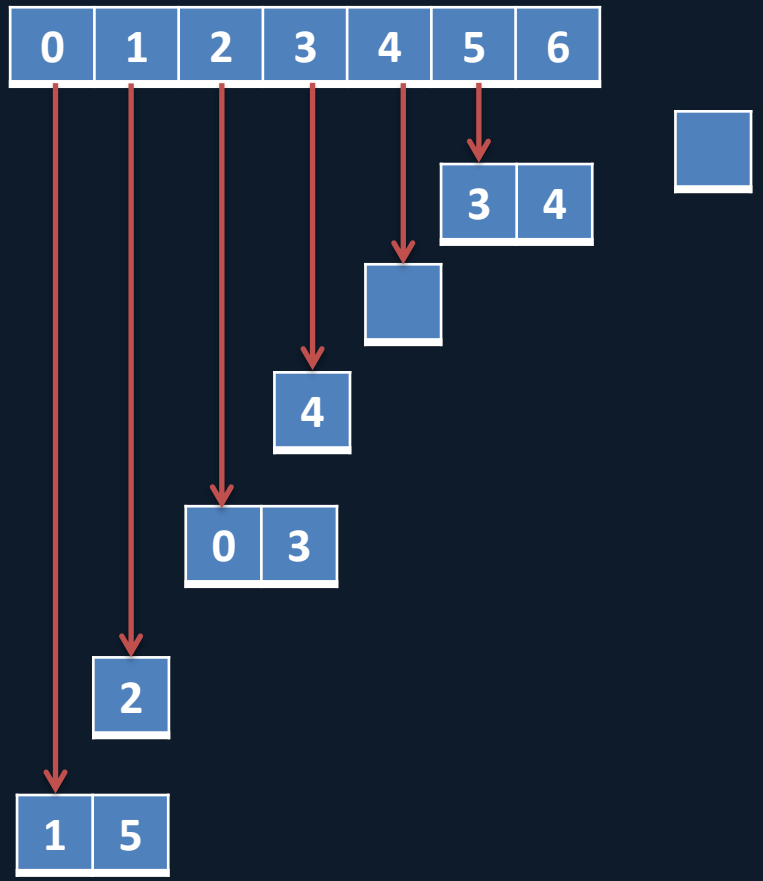


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Inserting an Edge

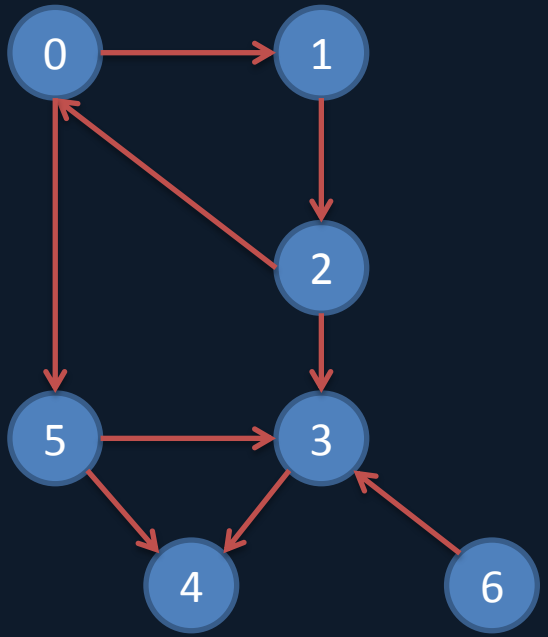


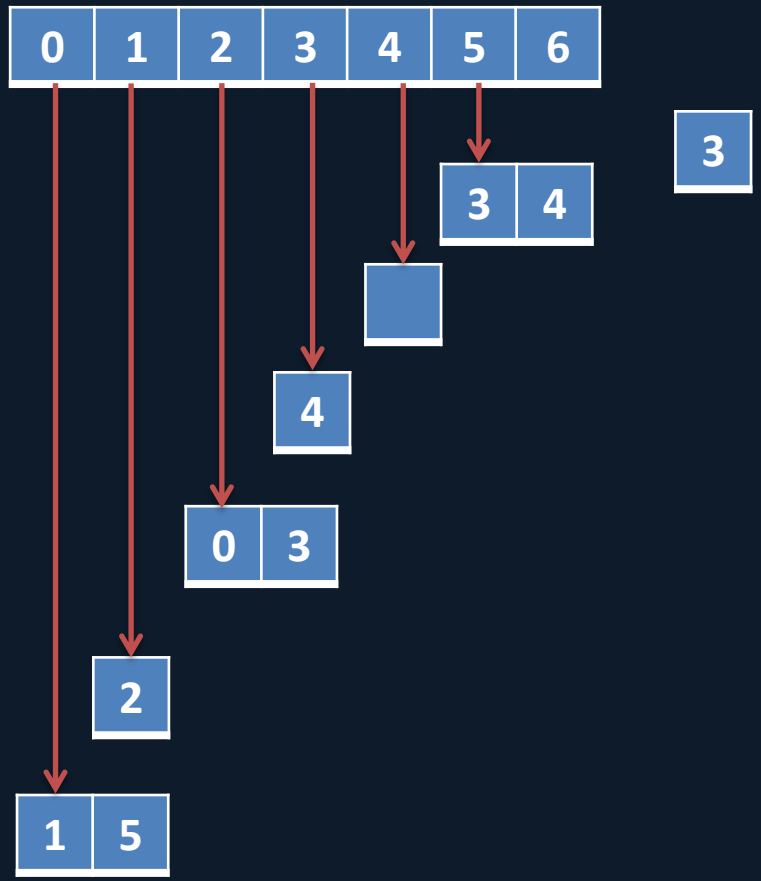


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Inserting an Edge

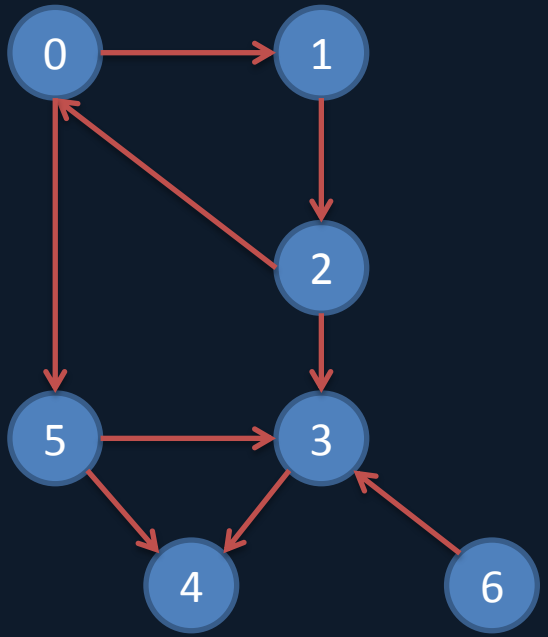


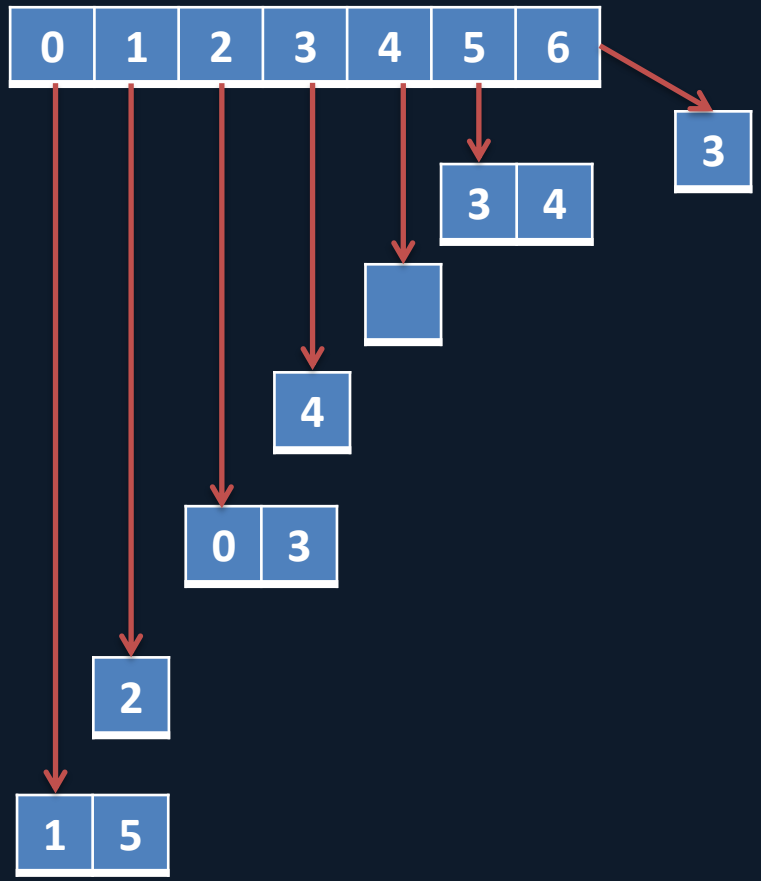


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Inserting an Edge





Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

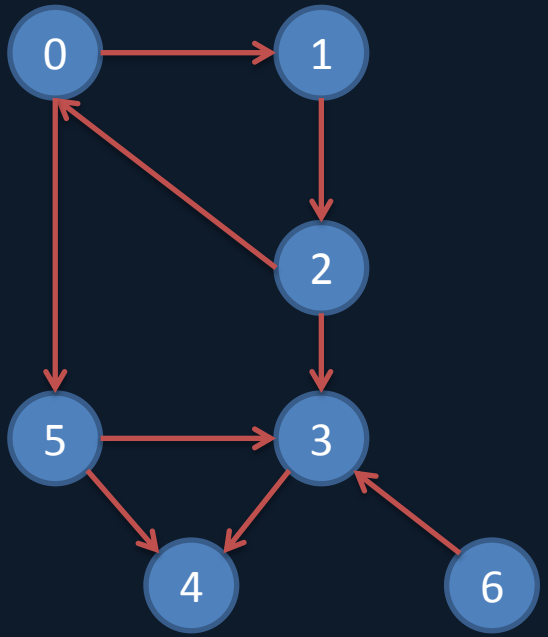
Graph Operations – Removing an Edge

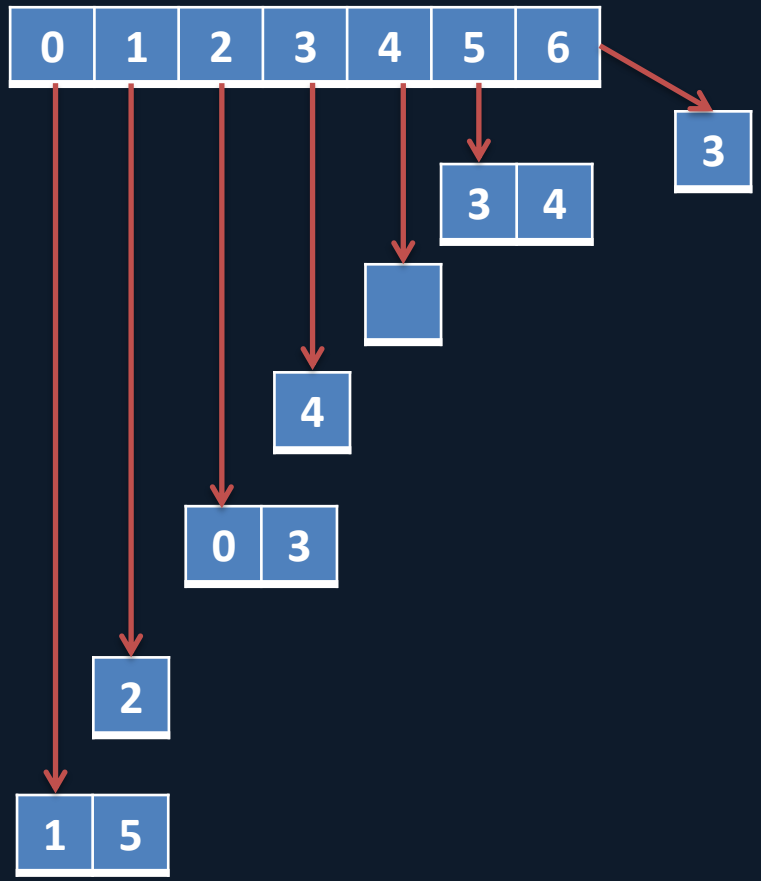
- To remove an edge going from node N1 to node N2:
  - We first loop through the edge list in N1, checking each edge's end pointer, checking if it is equal to N2.
  - Once we find the correct edge, we simply remove it from the edge list of the node.

Graph Operations – Removing a Node

- Removing a node is slightly more complicated as we need to ensure that all edges pointing to that node are first removed.
  - We first loop through all the nodes on our Graph.
  - For each node we loop through all of its edges.
  - For each edge, we check if it's end is equal to the node we want to remove.
  - If it is we remove it from the edge list for that node
  - Once we've looped through all the nodes in the graph we can remove the desired node from the node list.

Graph Operations – Removing Node 4

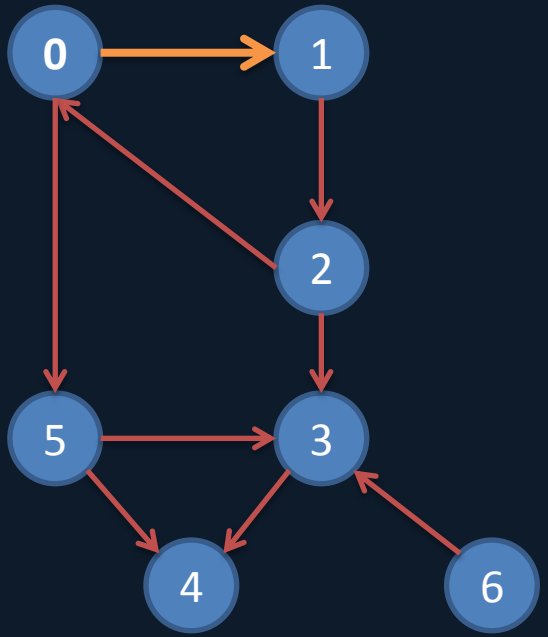


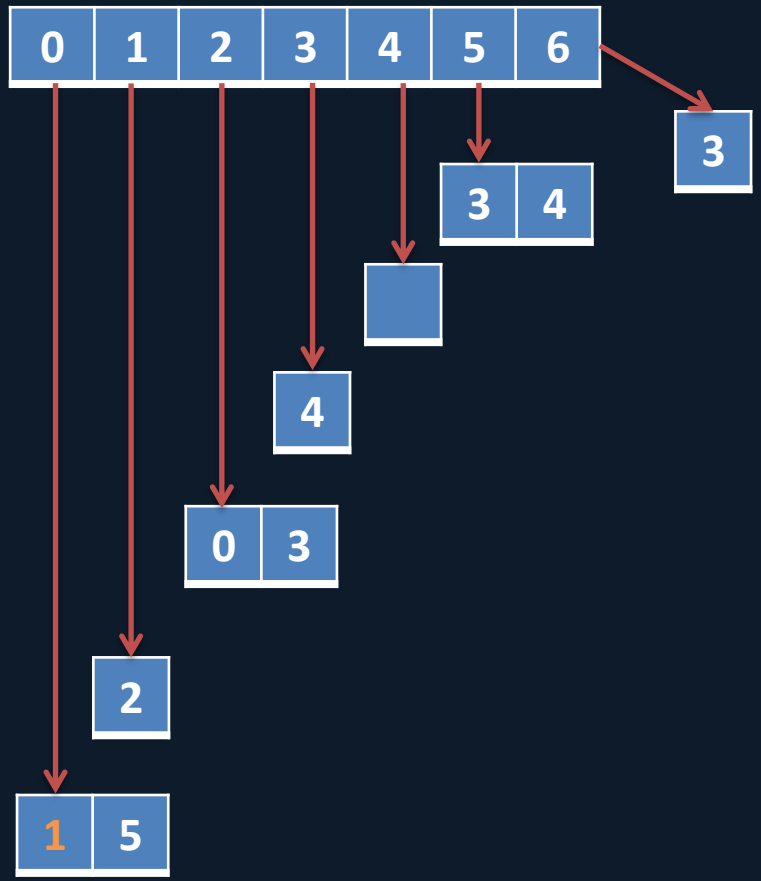


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

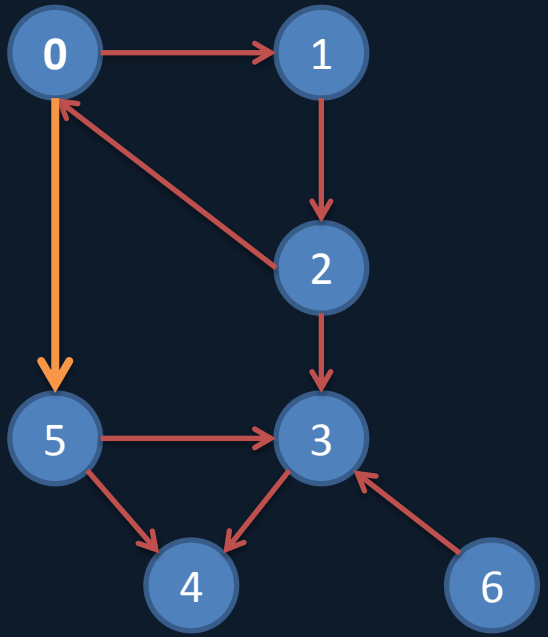


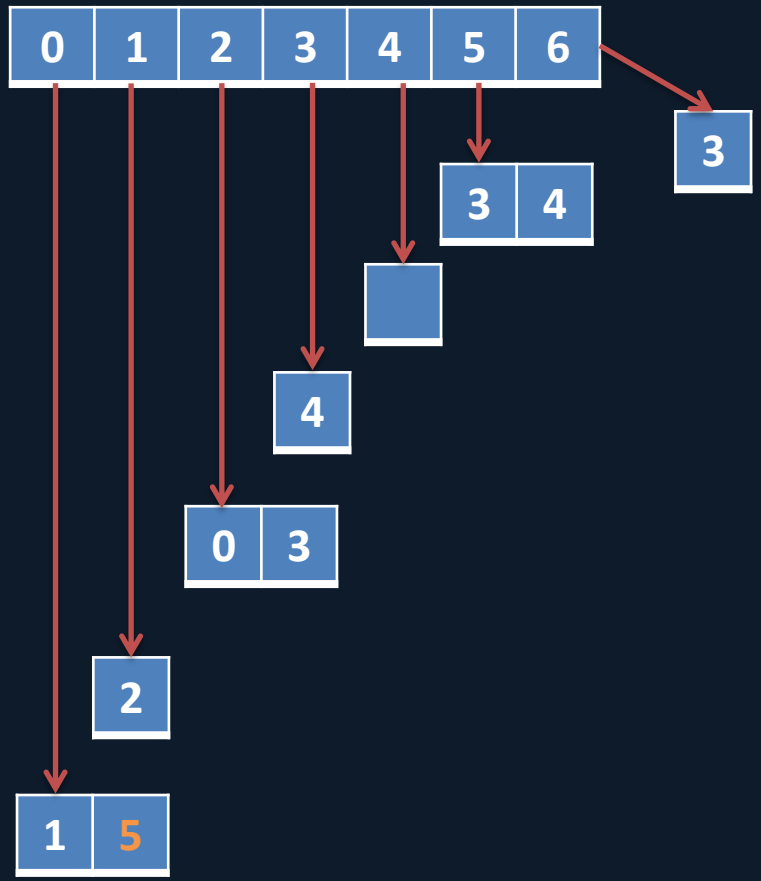


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

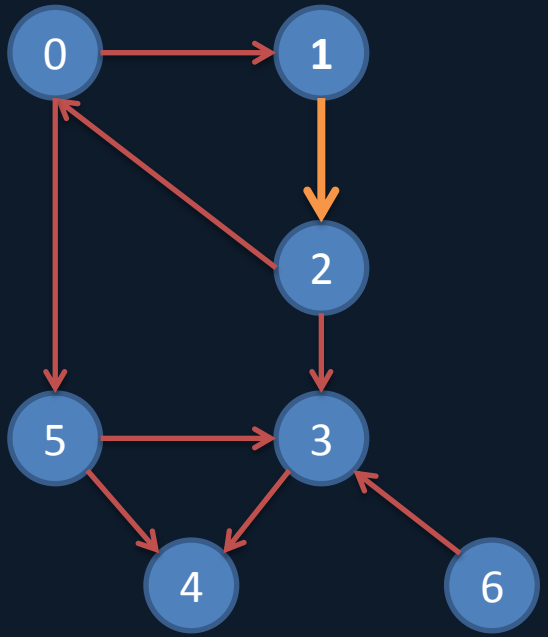


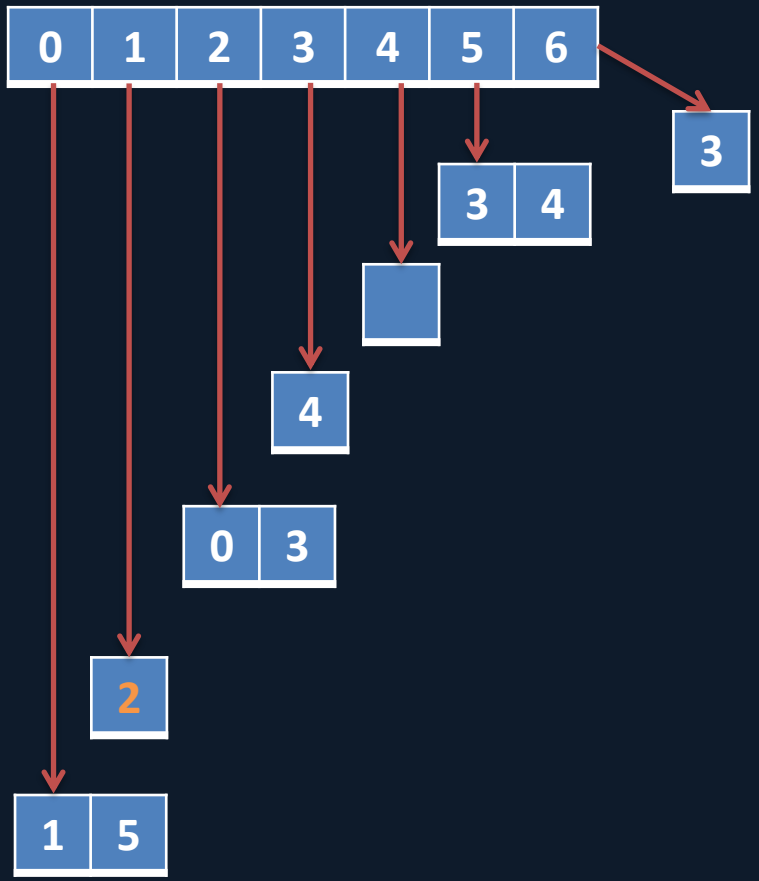


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

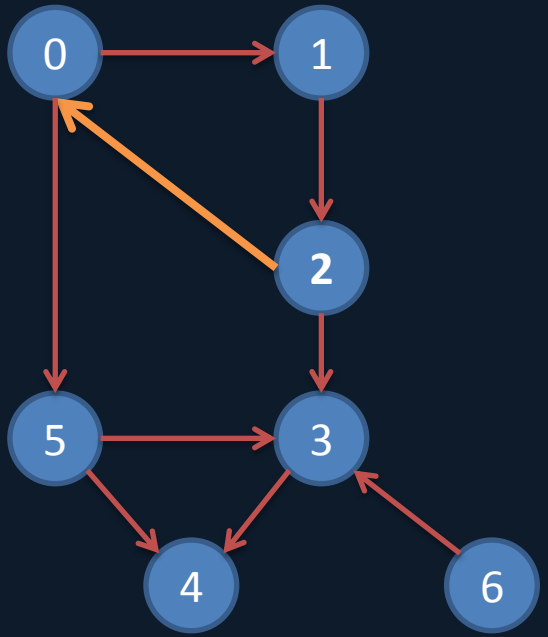


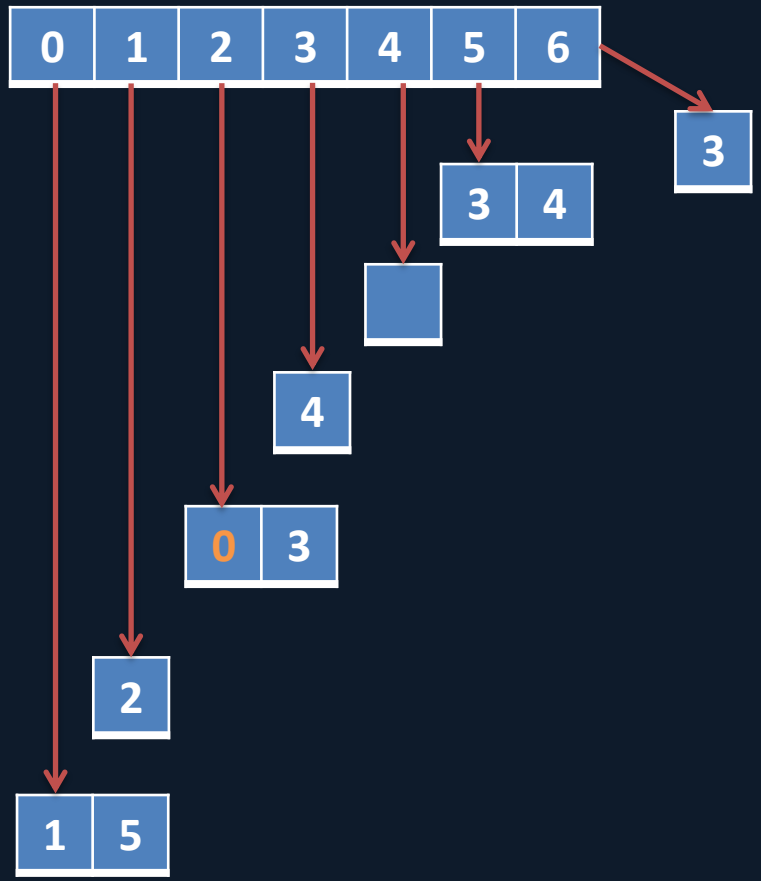


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

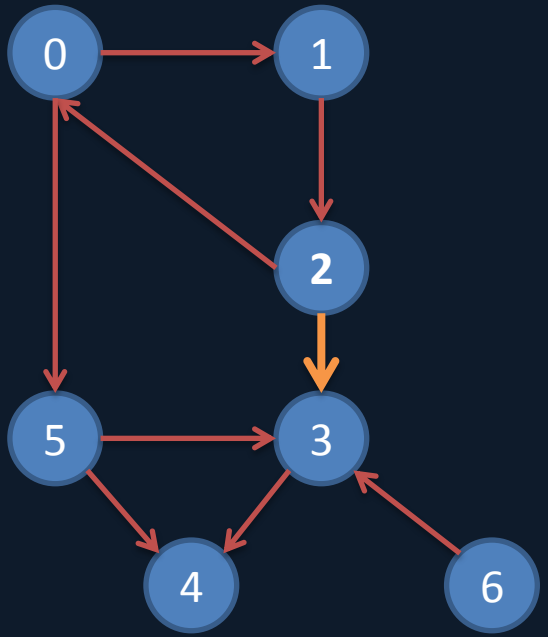


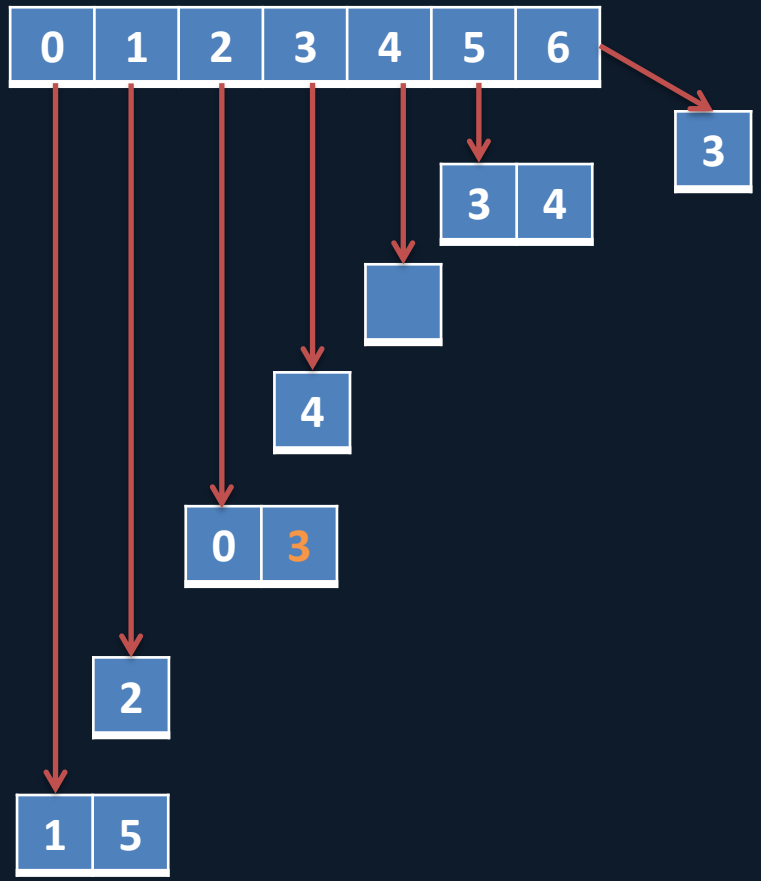


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

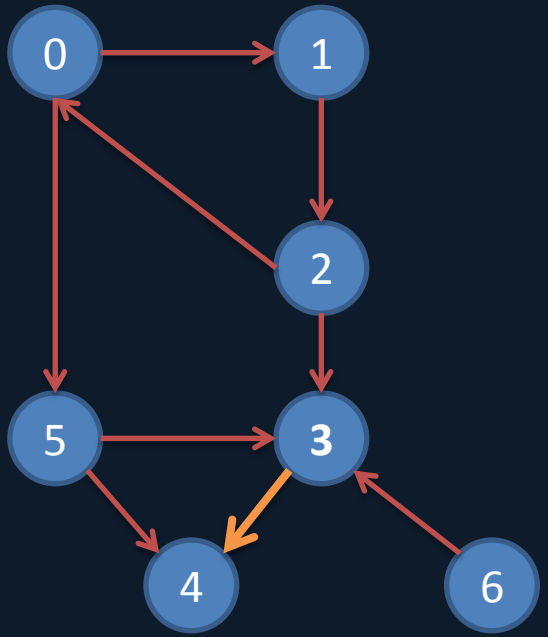


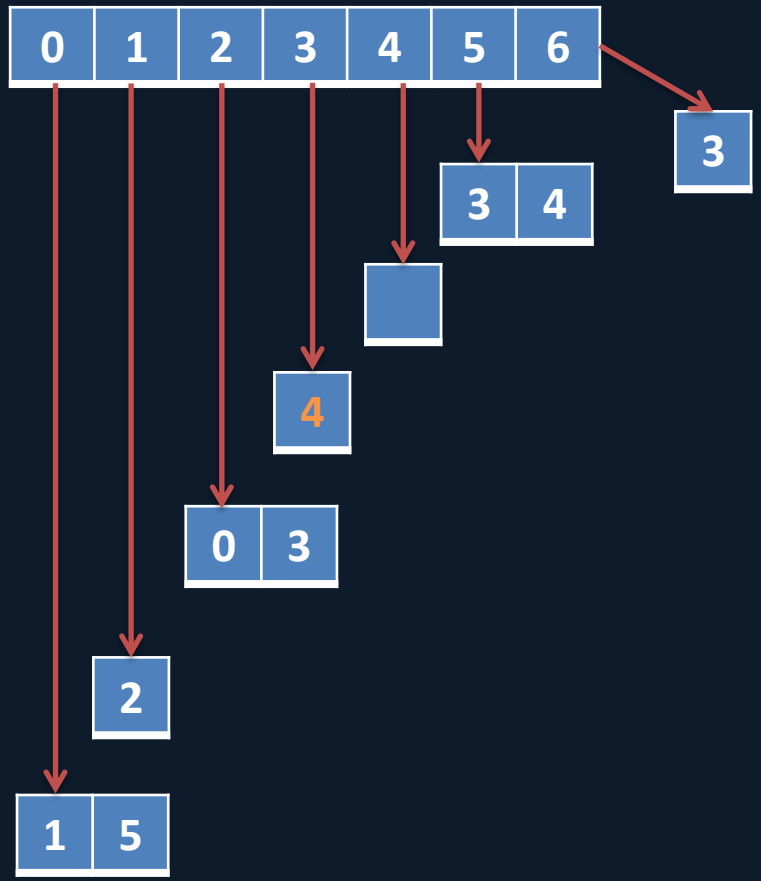


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

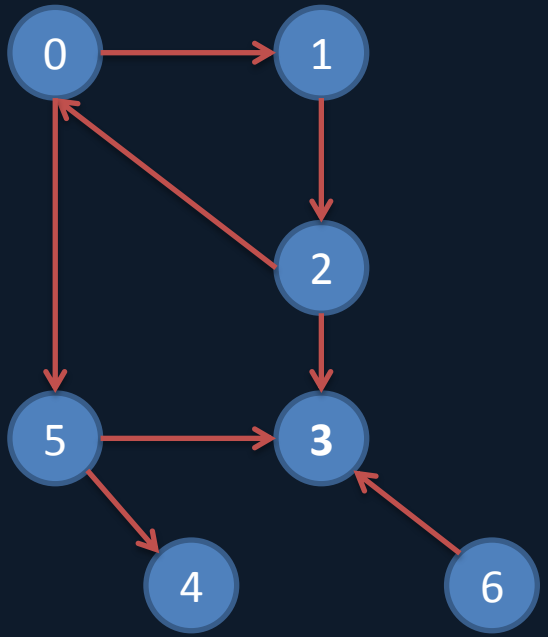


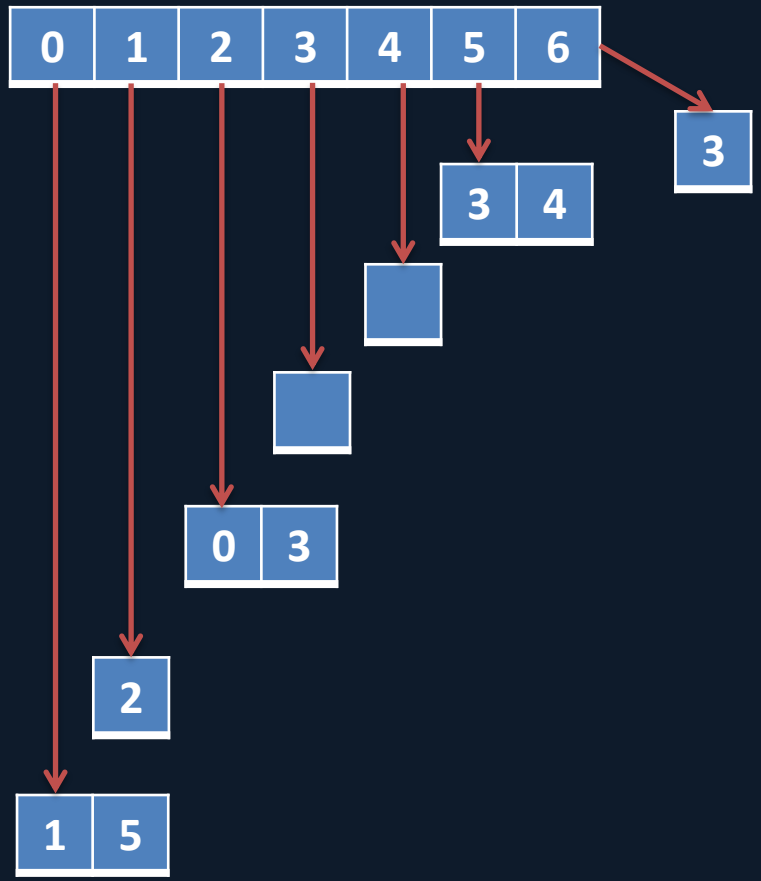


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

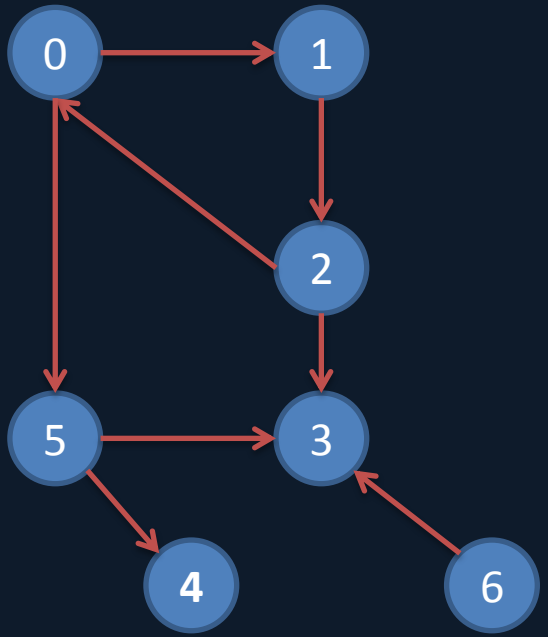


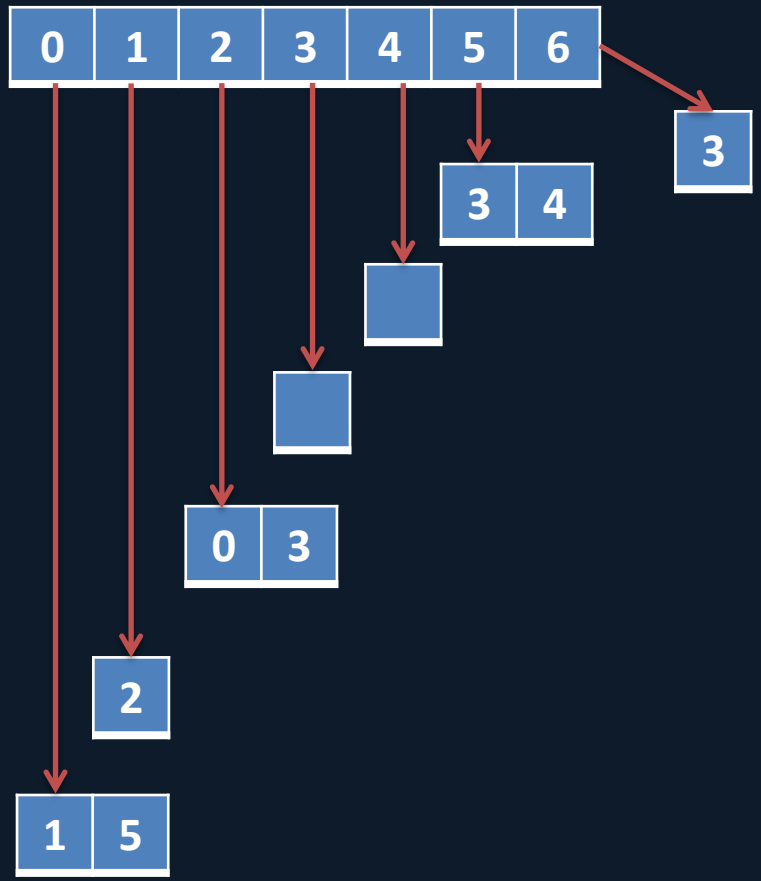


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

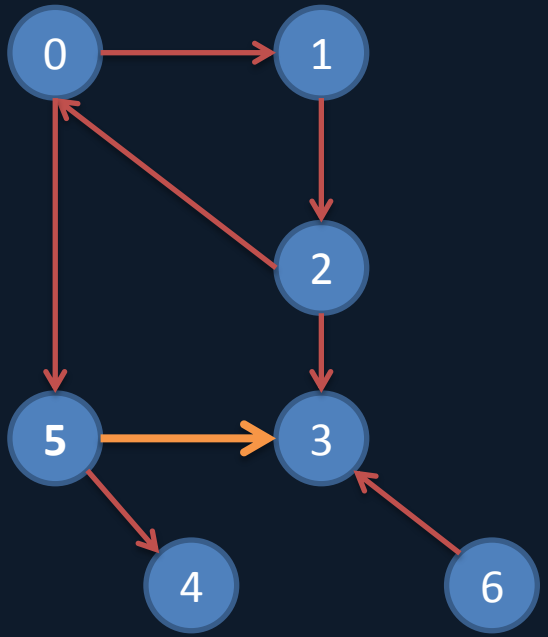


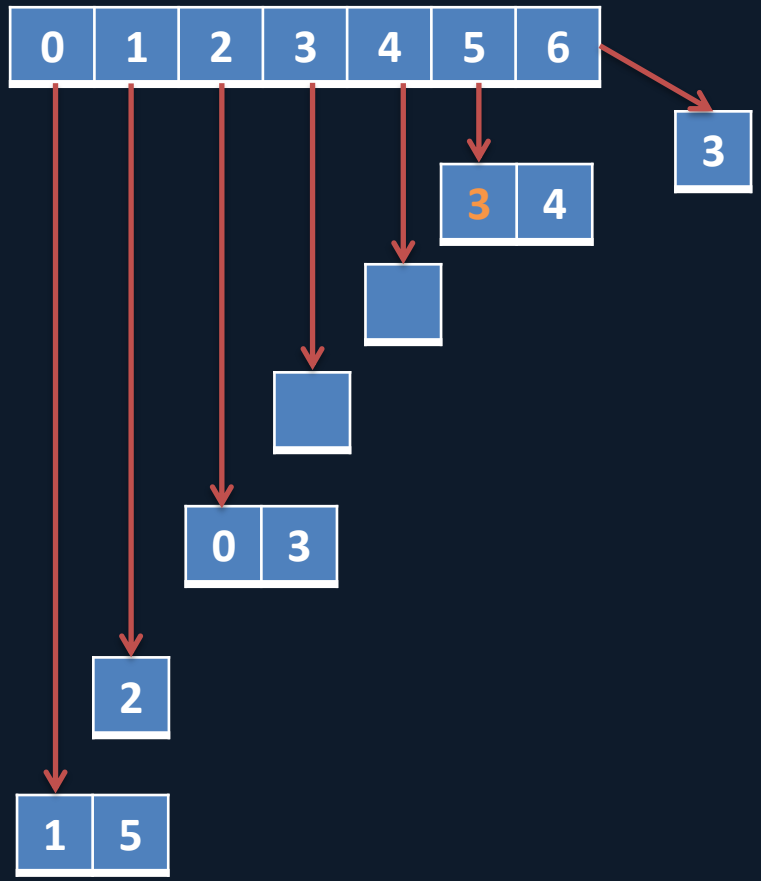


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

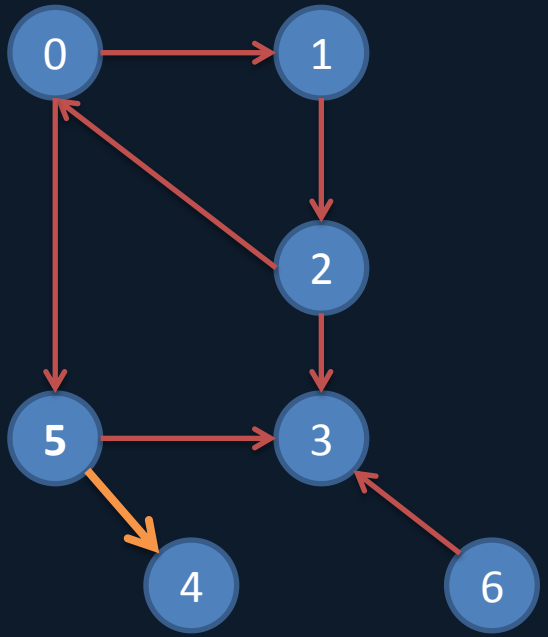


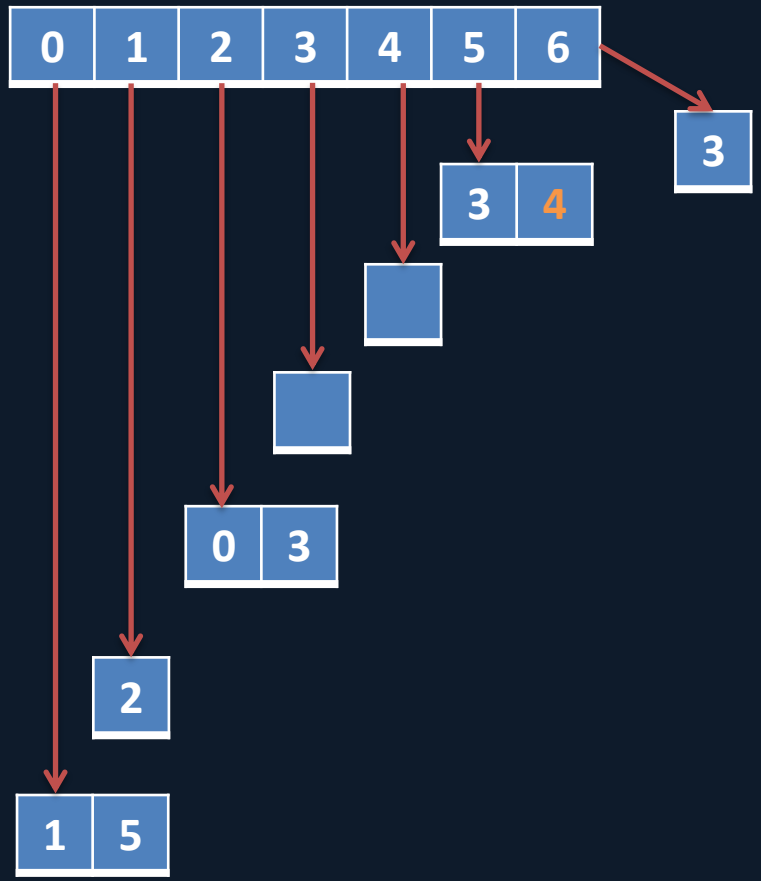


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

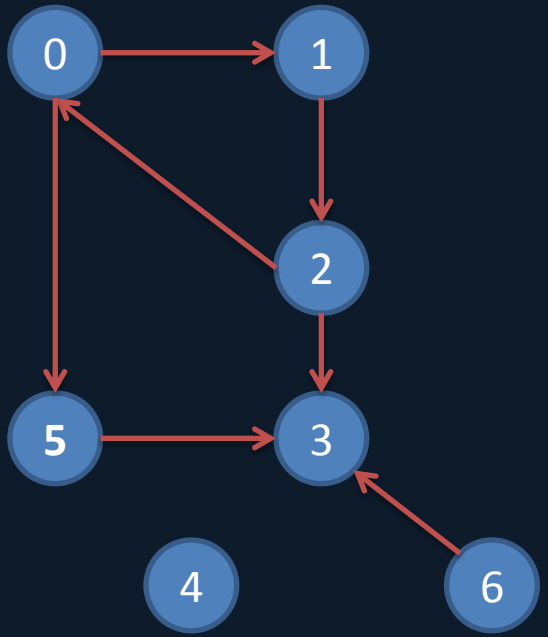


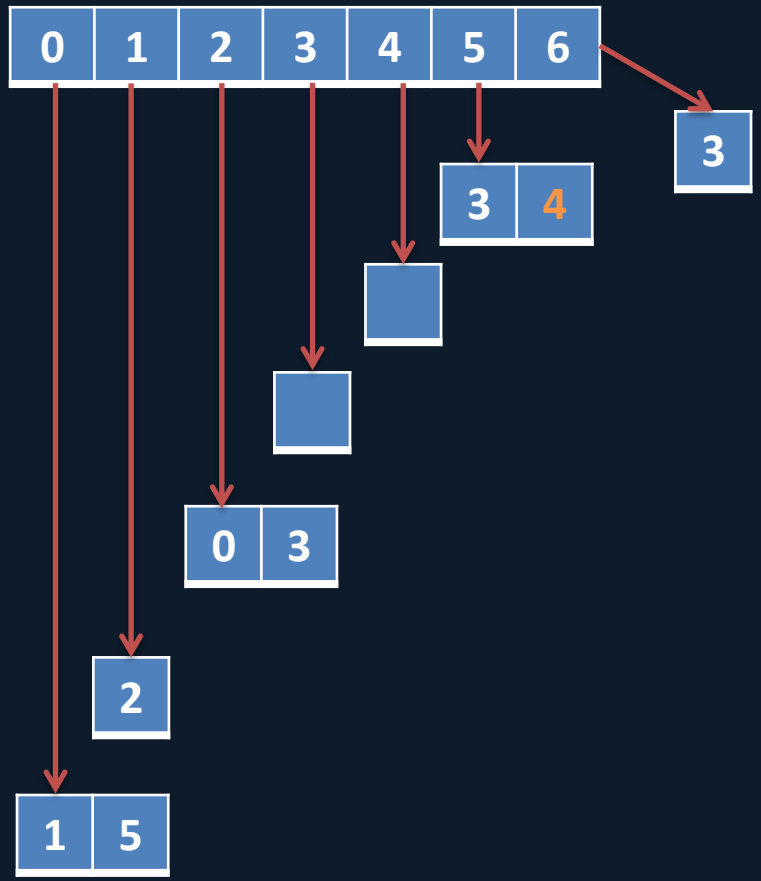


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

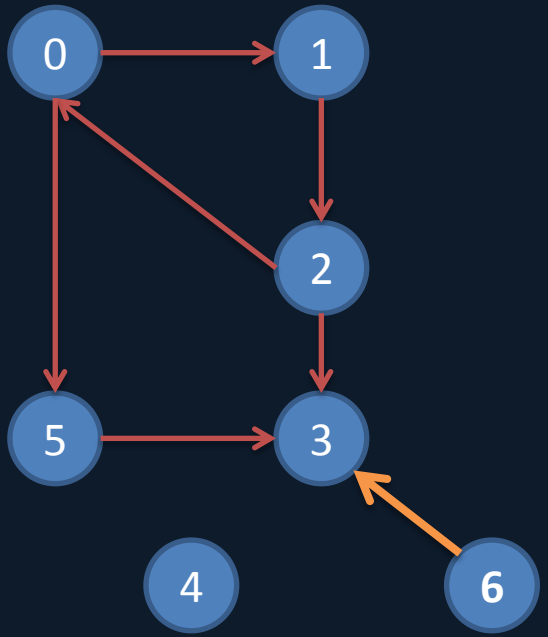


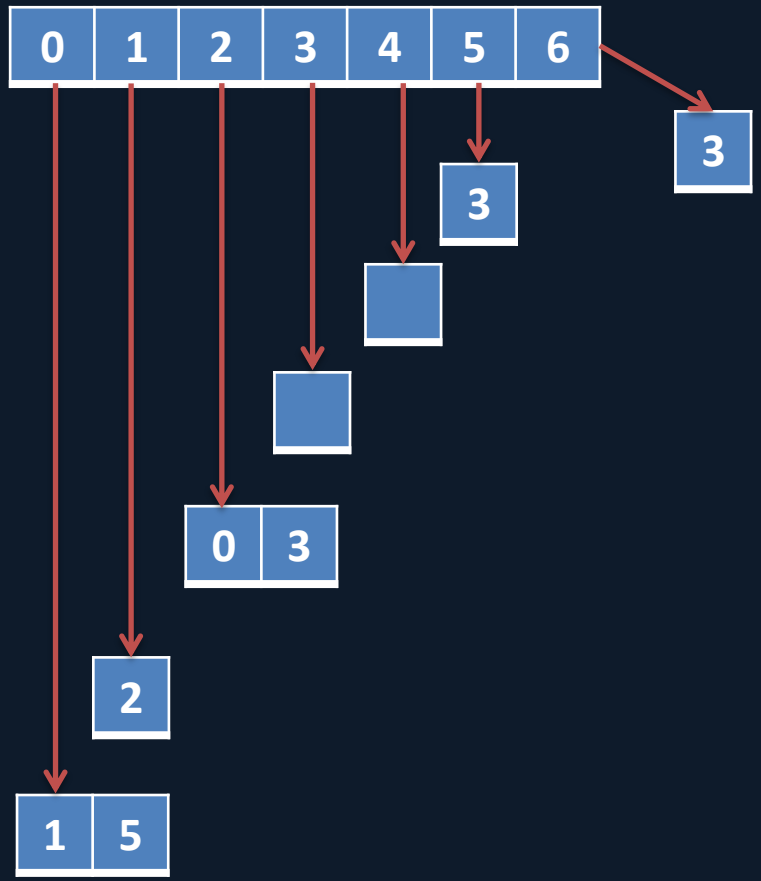


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

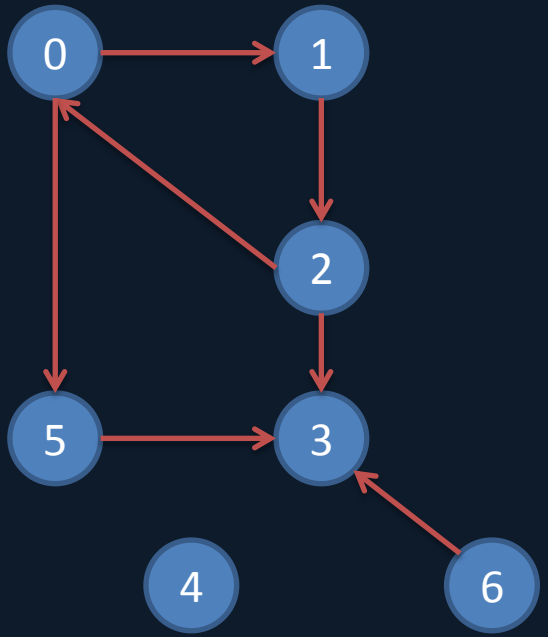


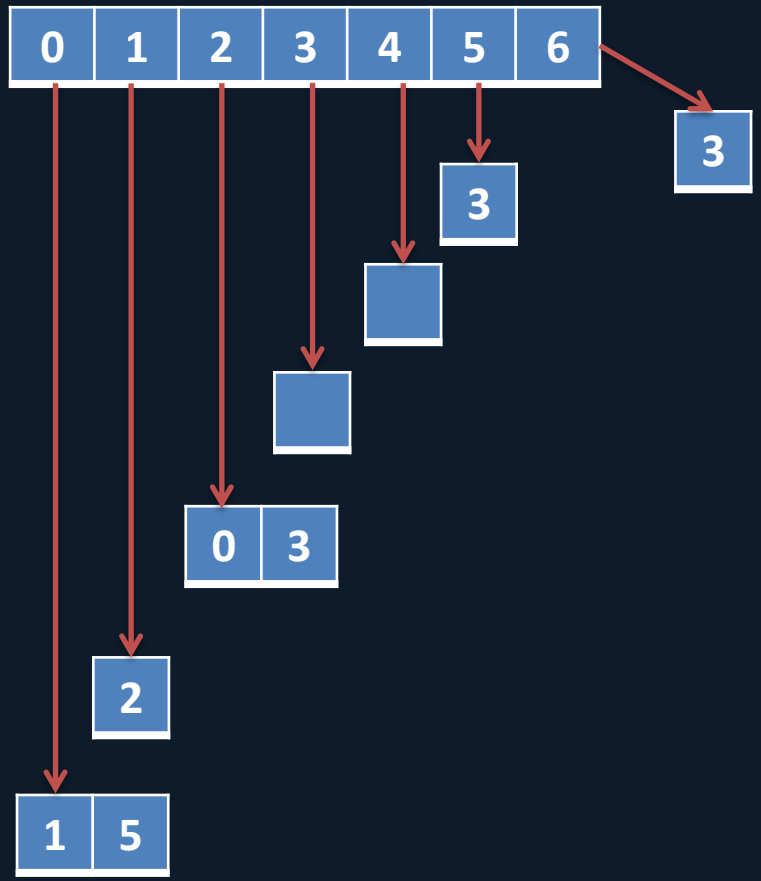


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4

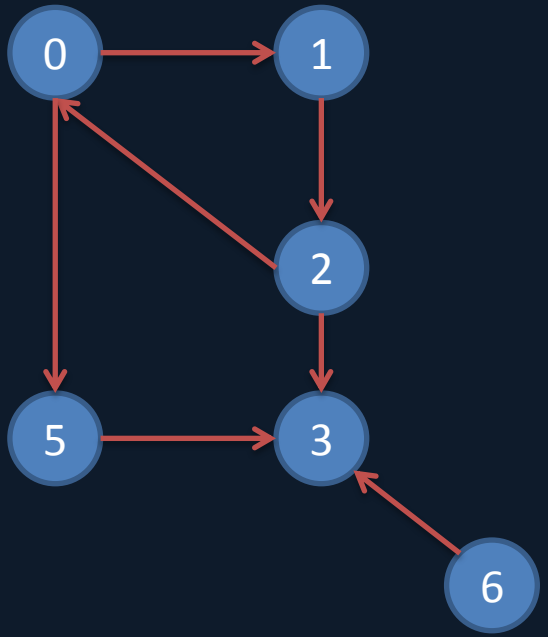


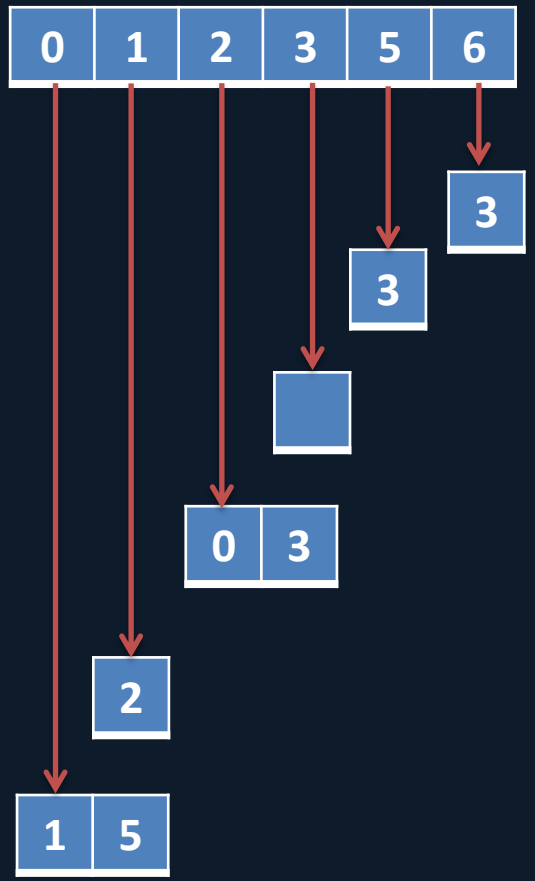


Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Graph Operations – Removing Node 4





Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Non-Directed Graphs

- To change these descriptions to suit a non-directed graph you only need to make a few changes.

- First, when adding or removing an edge, we need to add or remove it from the edge lists in both N1 and N2.

- Second, when removing a node, we only need to loop through the edges in the node we want to remove, erasing the edges from each end node.

Graphs in Games

- Graphs are used all over the place in games.

- Most commonly in path-finding, and decision making.
  - In pathfinding, nodes represent places you can walk, with edges being the paths between them.
    - Nodes can be walkable or not walkable,
    - edge costs can be just the distance between nodes or include extra costs such as walk speed penalties or danger
  - In decision making, nodes can be sequences of actions (such as attack or search for cover), and edges can be the transitions between behavioural states.
    - More advanced structures such as decision and behaviour trees exist and we cover them at the end of the topic.

- Also used for speeding up and organising rendering
  - Eg portal systems and scene graphs.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Questions?


Canberra Institute
of Technology


aie
SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX