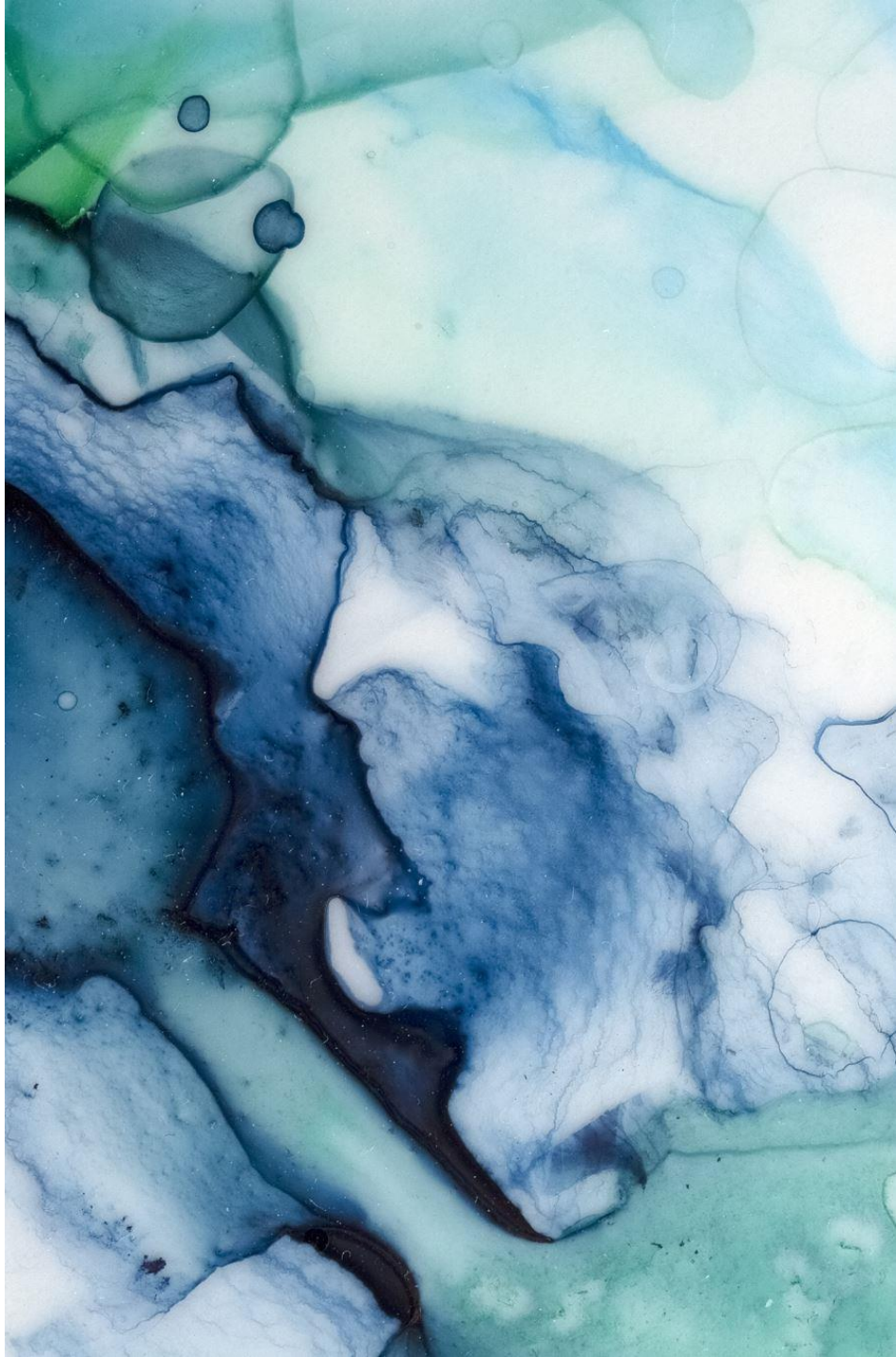
Cut vertices and separability

Team Members:

- Aatheeswaran K 106121004
- Appruval Kumar 106121018
- Sanjiv Kannaa J 106121116
- S Soma Vignesh 106121134

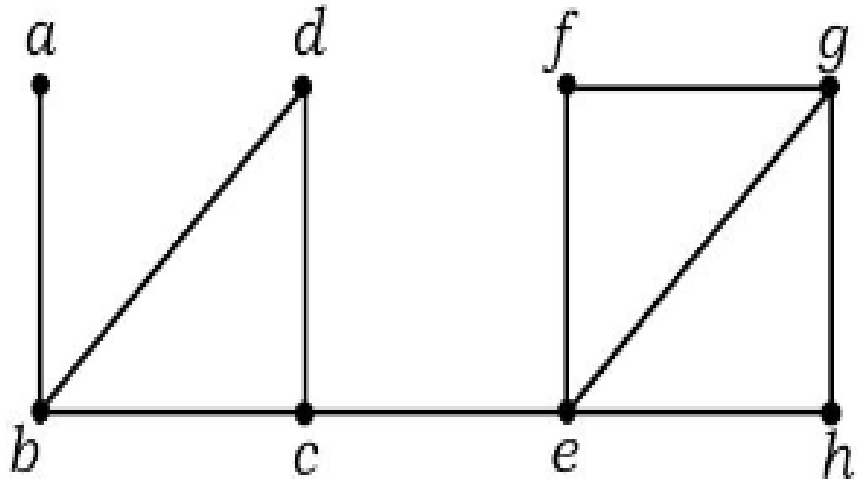


Cut Vertices

- A vertex in an undirected connected graph is an articulation point (or cut vertex) if removing it (and edges through it) disconnects the graph. Articulation points represent vulnerabilities in a connected network – single points whose failure would split the network into 2 or more components.
- They are useful for designing reliable networks. For a disconnected undirected graph, an articulation point is a vertex removal which increases the number of connected components.

Example of cut vertex

In this example, if either vertex c or e is removed the graph gets divided into 2 disconnected subgraphs, and hence is an example of a cut vertex



Cut vertices: c and e

Separability

- A graph is said to be separable if it is either disconnected or can be disconnected by removing one vertex, called articulation point or cut vertex.

- A graph that is not separable is said to be biconnected (or non-separable)

Adjacency Matrix

- Adjacency Matrix is a 2D array of size V x V where V is the number of vertices in a graph. Let the 2D array be adj[][], a slot adj[i][j] = 1 indicates that there is an edge from vertex i to vertex j.
- Adjacency matrix for undirected graph is always symmetric.
- Adjacency Matrix is also used to represent weighted graphs. If adj[i][j] = w, then there is an edge from vertex i to vertex j with weight w.

DFS Traversal

- *Depth-first search is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking.*
- *Start from the root or any arbitrary node and mark the node as visited and move to the adjacent unvisited node and continue this loop until there is no unvisited adjacent node. Then backtrack and check for other unvisited nodes and traverse them. Finally, print the nodes in the path.*

Algorithm to find the cut vertices and separability of undirected graph

A simple approach is to one by one remove all vertices and see if removal of a vertex causes disconnected graph.

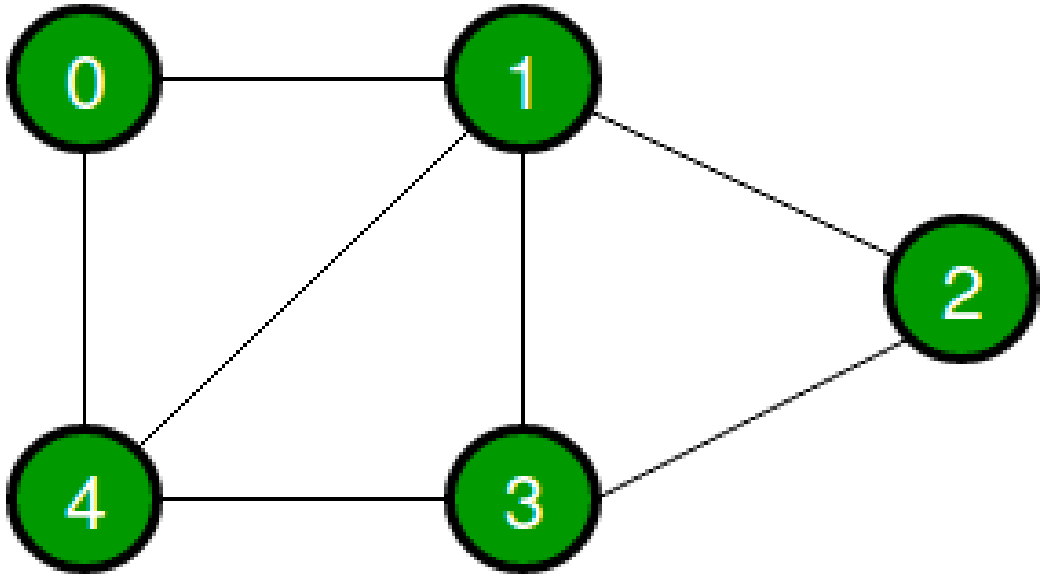
Iterate over all the vertices and for every vertex do the following:

- Remove the vertex from graph
- See if the graph remains connected (We use DFS traversal to check if the graph is still connected)
- Add the vertex back to the graph.

Time Complexity: O(V*(V+E)) for a graph represented using an adjacency list.

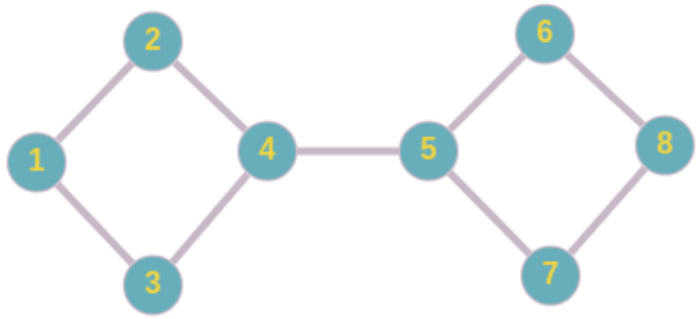
Auxiliary Space: O(V+E)

A graph and its adjacency matrix



	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	1	1
2	0	1	0	1	0
3	0	1	1	0	1
4	1	1	0	1	0

Sample Output




```
Enter the number of edges : 9
Enter the edge : 1 2
Enter the edge : 1 3
Enter the edge : 2 4
Enter the edge : 3 4
Enter the edge : 4 5
Enter the edge : 5 6
Enter the edge : 5 7
Enter the edge : 6 8
Enter the edge : 7 8

ADJACENCY MATRIX
0 1 1 0 0 0 0 0
1 0 0 1 0 0 0 0
1 0 0 1 0 0 0 0
0 1 1 0 1 0 0 0
0 0 0 1 0 1 1 0
0 0 0 0 1 0 0 1
0 0 0 0 1 0 0 1
0 0 0 0 0 1 1 0




4 is a cut vertex

5 is a cut vertex



it is a seperable graph
```