Bridges

Problem:

Given a connected undirected graph, you are asked to find all bridges in a graph. Bridges are edges without which the graph would get disconnected.

Solution:

Finding bridges can be solved in $O(n + m)$ using a DFS based approach. There are several algorithms. For example, Tarjan's Algorithm can be used.

Tarjan's Algorithm

We create two new arrays: $tin[u]$ storing the discovery time for each node u and $low[u]$ storing the lowest discovery time of all adjacent nodes except the parent node, for each node u. With this setup, we can say the following: If the lowest possible time to reach a vertex v is greater than discovery time of its parent u, so $low[v] > tin[u]$, it means that v can only be reached by its parent u. So if (u, v) is removed, v cannot be reached anymore. So (u, v) is a bridge.

Further reading: https://cp-algorithms.com/graph/bridge-searching.html

Note that Tarjan can be implemented recursively or iteratively. However, on very big graph instances (e.g. $n = 100000$, $m = 200000$), the recursive approach might introduce a segmentation fault in C++ by exceeding the default stack size limit of 1MB.