

## Determining if a given graph is a scorpion graph

Difficulty level: *high*

As discussed in the class, the use the adjacency matrix for storing a graph is usually impractical. This is because it usually leads to an extra overhead in the time and space complexity for solving an algorithmic problem. Firstly, the graphs in real applications are sparse, and so the space  $O(n^2)$  used by adjacency matrix is too large to be stored in RAM. Secondly, if the algorithm requires enumerating and processing all neighbors of each vertex, then it will require  $O(n^2)$  overhead in the time complexity as well. However, there are some nice problems which require only  $O(n)$  time complexity if we use adjacency matrix. One such problem that we discussed in the class was the problem of determining if a given directed graph has a sink. In this gem problem, we are going to discuss another nice (but much more difficult) problem which requires  $O(n)$  time complexity if we are given adjacency matrix representation.

An  $n$ -vertex graph is a scorpion if it has a vertex of degree 1 (the **sting**) connected to a vertex of degree two (the **tail**) connected to a vertex of degree  $n - 2$  (the **body**) connected to the other  $n - 3$  (the **feet**). Some of the feet may be connected to other feet. Design an algorithm that decides whether a given adjacency matrix represents a scorpion by examining only  $O(n)$  entries. An example of scorio is given in Figure 1 below.

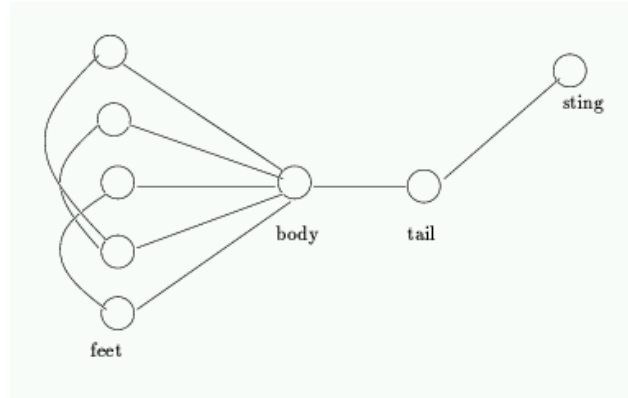


Figure 1: A scorpion graph