This question has been translated from Standard ML to OCaml

A rooted directed graph has vertices identified by integers. Each vertex \( v \) has a left successor given by \( \text{left } v \) and a right successor given by \( \text{right } v \), where \( \text{left} \) and \( \text{right} \) are OCaml functions of type \( \text{int} -> \text{int} \). The graph contains the root and all vertices reachable by paths from the root.

Define a variant type \( \text{graph} \) that could be used to represent such a graph with given root, and left and right functions, and define a function \( \text{mkgraph root left right} \) that can create values of type \( \text{graph} \). Show that such values can be used to represent both finite and infinite graphs.

A path through the graph is represented by a \( \text{bool list} \) with \( \text{true} \) and \( \text{false} \) indicating left and right edges, respectively.

Define the function \( \text{last : graph -> bool list -> int} \) that will yield, for a given graph, the identity of the vertex reached by following the given path from the root.

In a new application, where \( \text{last} \) is repeatedly called, it is required for it to return both the identity of the last vertex and a count of how often this particular vertex has been returned. Define a new version of the variant type \( \text{graph} \), containing mutable values, that could be used.

Illustrate the use of this type by defining the new versions of \( \text{mkgraph} \) and \( \text{last} \).