

We label the vertices  $v_1, v_2, \dots, v_n$  according to a topological ordering. We now define  $Win(j)$  to be equal to 1 if the player whose turn it is to move can force a win starting at node  $v_j$ , and define  $Win(j)$  to be equal to 0 if the other (who isn't about to move) can force a win starting at node  $v_j$ .

We can initialize  $Win(j) = 0$  for every node  $v_j$  with no out-going edges. In particular, this means that we will set  $Win(n) = 0$ . We now use dynamic programming to compute the values of  $Win(j)$  in descending order of  $j$ . When we get to a particular value of  $j$ , we may assume that we have already computed  $Win(k)$  for all  $k > j$ . Now, a player starting from  $v_j$  can force a win if and only if there is some node  $v_k$  for which  $(v_j, v_k)$  is an edge and a player starting from  $v_k$  has a forced loss. Thus,  $Win(j) = 1$  if and only if  $Win(k) = 0$  for some  $k$  with  $(v_j, v_k)$  an edge; and otherwise  $Win(j) = 0$ .

We thus compute all these values in  $O(n)$  time per entry, for a total of  $O(n^2)$ . We then simply check the value of  $Win(j)$  for the node  $v_j$  on which the game is designated to start.