 **Pseudo Code**

In the following algorithm, the code has two functions. First, building the graph of the input term and find the meanings of the term, and vice versa. Until the depth count to zero, the graph will be generated.

|  |
| --- |
| Algorithm BuidlingTree is  Input: Word : word that you search about  maxDepth : number of times for reading from dictionary, default=10  prev : previous node in graph, default=""    if maxDepth greater than ZERO  define meanings  if word is english  get all arabic meanings for it, then store it in 'meanings'  else  get all english meanings for it, then store it in 'meanings'  end if    /\* loop through meanings \*/  for each meaningWord in meanings  build new node 'newNode' for meaningWord  create link goes from node to newNode  create link goes from newNode to node  call recursive function BuidlingTree with inputs (newNode, maxDepth -1, node)  end for    end if  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  Function FindCircleGraph is  input : node : first node in the tree  Intial word  stack : Empty Stack  result : a refernce to result  /\* check if the node has more than one links \*/  if node.links >= 2  /\* if item found in the stack for second time, then a circle is found \*/  if (stack contains current node) and (current node equal the first node)  copy all items from stack into result because all nodes in the stack form the circle  endif    if not internal circle is found  add current node to the stack  for each link in current node links  recursive call for FindCircleGraph with parameters(link, initial word, stack, result)    end for  remove the current node from the stack  endif;    endif; |