```
function meta_computation():
//Compute the degree and parent id for each vertex

function build_inconsistency_list (UpdateList edge_list
,Graph G, Priority MIN_PRIORITY):
min_depth = MAX_INT
for each e in edge_list
    for each in-edge e' with e'.dst = e. dst
        min_depth = min(VertexProperty[e'.src].depth
       ,min_depth)
    current_depth = VertexProperty[e.dest].depth
    if(current_depth > min_depth + 1)
        VertexState[e.dst].depth =  min_depth + 1
        inconsistency_list =
        inconsistency_list U {e.dst,MIN_PRIORITY_QUEUE}
G' = G
return (inconsistency_list,G')

function property_guard(DepthThreshold DP
,ThresholdFraction f):
if fraction of inconsistent vertices
         with (depth < DP) > f
  run static re-computation
else run incremental algorithm

function frontier_activate(Graph G'
,List inconsistency_list):
//Extract all duplicate minimums
v_min_set = inconsistency_list.extract_min()
Activate(v_min_set)

function update_inconsistency_list(Graph G'
,List inconsistency_list,Set frontier):
//Similar to build_inconsistency_list() but checks for
//consistency of all the successors of the frontier

//I-GAS computation loop
function I-GAS(List inconsistency_list,Graph G'):
while(!inconsistency_list.empty())
  frontier = frontier_activate(G',inconsistency_list)
  IGAS(G')
  update_inconsistency_list(G',inconsistency_list
  ,frontier)

function merge_state(Graph G,Graph G')
// Do nothing in BFS
```