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

function build_inconsistency_list(edge_list,
G, DEFAULT_PRIORITY):
//For each edge inserted, add an edge to G'
//if endpoints belong to different components
for each edge e in edge_list
  label1 = VertexProperty[e.src].component_label
  label2 = VertexProperty[e.dst].component_label
  if(label1 != label2)
    G' = G' U (label1, label2)
    inconsistency_list=
    inconsistency_list U {e.src,e.dst,DEFAULT_PRIORITY}
return (inconsistency_list,G')

function property_guard(degree: DE,
disjoint component: count threshold_fraction f):
    if fraction of inconsistent vertices with
    (degree > DE or count > n*f) > f
       run static re-computation
    else run incremental algorithm

function frontier_activate(G',inconsistency_list):
// Using extract operation on inconsistency_list
for every edge e in G'
  activate(e.src)
  activate(e.dst)

function update_inconsistency_list(G',
inconsistency_list, new_inconsistency=NULL):
if(G.activity.empty())
   inconsistency_list.clear()

//I-GAS computation loop
function I-GAS(inconsistency_list,G'):
While(!inconsistency_list.empty())
  if(itr=1)
     frontier_activate(G',inconsistency_list)
  else frontier_activate(G',NULL)
  Modified-GAS(G')
  update_inconsistency_list(G',inconsistency_list)

function merge_state():
 // relabel the vertex component ids
```