Algorithm 1 The pseudocode for Evaluate

1: num = 0, push(root, S)
2: DFN[root] = num; LOW[root] = num; ST_SIZE[root] = 1
3: VISITED[root] = true; num = num + 1
4: repeat
5:   Vertex v = peek(S)
6:   y = next unvisited neighbor of v
7:   if y exists then
8:     mark y as visited, push(y, S)
9:     DFN[y] = num; LOW[y] = DFN[y]; PARENT[y] = v;
10:    ST_SIZE[y] = 1; IMPACT[y] = 0
11:   else
12:      pop(S)
13:      for all neighbors w of v do
14:         if DFN[w] < DFN[v} and PARENT[v] ≠ w then
15:             LOW[v] = min(LOW[v], DFN[w])
16:         else if PARENT[w] == v then
17:             LOW[v] = min(LOW[v], LOW[w])
18:             if !COUNTED[w] and (PARENT[v] ≠ w or v is root) then
19:                 COUNTED[w] = true
21:             end if
22:             if LOW[w] ≥ DFN[v] and v ≠ root then
23:                 mark v as an articulation point
25:                 IMPACT[v] = IMPACT[v] + f(ST_SIZE[w])
26:             end if
27:         end if
28:      end for
29:   end if
30: until Stack is empty
31: for each visited and counted vertex v do
32:   if v is an articulation point then
33:     IMPACT[v] = IMPACT[v] + f(num - CUT_SIZE[v])
34:   else
35:     IMPACT[v] = IMPACT[v] + f(num - 1)
36:   end if
37: end for
38: Maintain the vertex v* with the minimum IMPACT value
39: return v*