Algorithm 1: Distributed privacy preserving sub-graph mining

Data: Each client maintains its local graph LG_i, all LG_i constitute a global graph GG, each node n_i^m in LG_i has its type t_i^m, there are N_t types in total, each directed edge also has its relation whether it belongs to LG_i or across LG_i and LG_j, all relation in GG forms collection R.

Result: legal sub-gragh of GG with K edges

```
1  Initial a multitree T with an empty root;
2  Genarate N_t TreeNodes with unique type at the next level in T;
3  depth ← 0 ;
4  for depth <= K do
       /* Expand as much as possible                                  */
5      foreach TreeNode TN_i at depth^th level of T do
6          Graph g ← content of TN_i;
7          foreach node SouN_x in g do
8              foreach relation r_p in R do
9                  foreach node TarN_y with unique type do
10                     Genarate graph pattern (SouN_x, r_p, TarN_y);
11                     if g + (SouN_x, r_p, TarN_y) is legal judged by
                        Algorithm2 then
12                         g_new ← g + pattern (SouN_x, r_p, TarN_y);
13                         A child of TN_i ← TN_ij with content g_new ;
14                     else
15                         continue;
16                     end
17                 end
18             end
19         end
20     end
       /* duplicate removal                                          */
21     for every different (TN_i, TN_j) pairs at depth^th level of T do
22         if TN_i <==> TN_j then
23             Remove TN_j from T;
24         end
25     end
26     depth ← depth + 1 ;
27 end
```