Algorithm: All Maximimal Cliques
(Tsukiyama, Ide, Ariyoshi, and Shirakawa, 1977)

```
input : undirected graph G = (V = {1,...,n}, E)
output: all maximal cliques of G

maximal( vertex set K, vertex i) begin
    // is K maximal clique in G[1...i] ?
    for j = 1,...,i do
        if j ∉ K and K ⊆ N(j) then return false
    return true

parent( vertex set K, vertex i) begin
    // lexicographically first maximal clique
    // in G[1...i-1] that contains K-i
    P ← K \ {i}
    for j = 1,...,i-1 do
        if P ⊆ N(j) then P ← P ∪ {j}
    return P

insert( vertex i) begin
    if i = n+1 then
        list maximal clique C
    else
        if C ⊆ N(i) then
            // only child
            C ← C ∪ {i}
            insert(i+1)
            C ← C \ {i}
        else
            // left child
            insert(i+1)

            // right child
            K := (C ∩ N(i)) ∪ {i}
            if maximal(K,i) and C = parent(K,i) then
                C ← K
                insert(i+1)
                C ← parent(C,i)

begin
    C ← ∅
    insert(1)
```

Remark: $N(i)$ defines the neighbours of vertex i in the graph.