Input : undirected graph $G = (V, E)$.
Output : vertex ordering σ.

```
1  assign each vertex label ∅;
2  for i ← n to 1 do
3  |  choose a vertex v
   |        with no assigned number in σ
   |              with lexicographically largest label;
4  |  σ(i) ← v;
5  |  for every vertex w ∈ Adj(v)
   |  |        with no assign number in σ
6  |  |  append i to label(w);
7  |  end for
8  end for
```

Algorithm 1 : LexBFS

```
1  for w ∈ Adj(v) not numbered do
2  |  if Flag(Set(w)) = false then
3  |  |  insert new set S before Set(w) into Q;
4  |  |  Flag(Set(w)) ← true; add Set(w) to FixList;
5  |  end if
6  |  S ← set before Set(w) in Q;
7  |  remove w from Set(w); add w to S;
8  |  Set(w) ← S;
9  end for
10 for S ∈ FixList do
11 |  Flag(S) ← false;
12 |  if S empty then
13 |  |  remove S from Q;
14 |  end if
15 |  remove S from FixList;
16 end for
```

Algorithm 2 : Update step in LexBFS

Input : graph $G = (V, E)$, vertex ordering σ.
Output : true, if σ PES, false otherwise.

```
1  for each vertex v do A(v) ← ∅;
2  for i ← 1 to n − 1 do
3  |  v ← σ(i);
4  |  X ← {x ∈ Adj(v) | σ(v) < σ(x)};
5  |  if X = ∅ then go to line 8;
6  |  u ← argmin{σ(x) | x ∈ X};
7  |  add X − {u} to A(u);
8  |  if A(v) − Adj(v) ≠ ∅ then
9  |  |  return false;
10 |  end if
11 end for
12 return true;
```

Algorithm 3 : Test for perfect elimination scheme

Input : lists $\mathrm{Adj}(v)$, $A(v)$.
Output : true, if $A(v) - \mathrm{Adj}(v) \neq \emptyset$, false otherwise.

```
1  for w ∈ Adj(v) do Test(w) ← true;
2  for w ∈ A(v) do
3  |  if Test(w) = false then
4  |  |  return true;
5  |  end if
6  end for
7  for w ∈ Adj(v) do Test(w) ← false;
8  return false;
```

Algorithm 4 : Test for $A(v) - \mathrm{Adj}(v) \neq \emptyset$ in line **8**