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

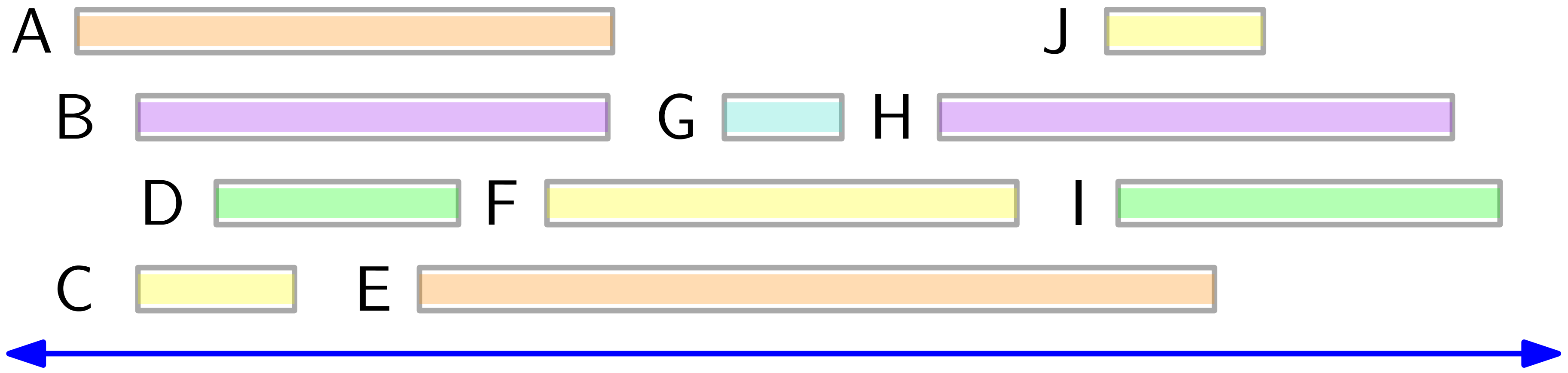
```
1  for each vertex v do A(v) ← ∅;
2  for i ← 1 to n − 1 do
3  │  v ← σ(i);
4  │  X ← {x ∈ Adj(v) | σ(v) < σ(x)};
5  │  if X = ∅ then go to line 8;
6  │  u ← argmin{σ(x) | x ∈ X};
7  │  add X − {u} to A(u);
8  │  if A(v) − Adj(v) ≠ ∅ then
9  │  │  return false;
10 │  end if
11 end for
12 return true;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

Algorithm 3 : Test for perfect elimination scheme

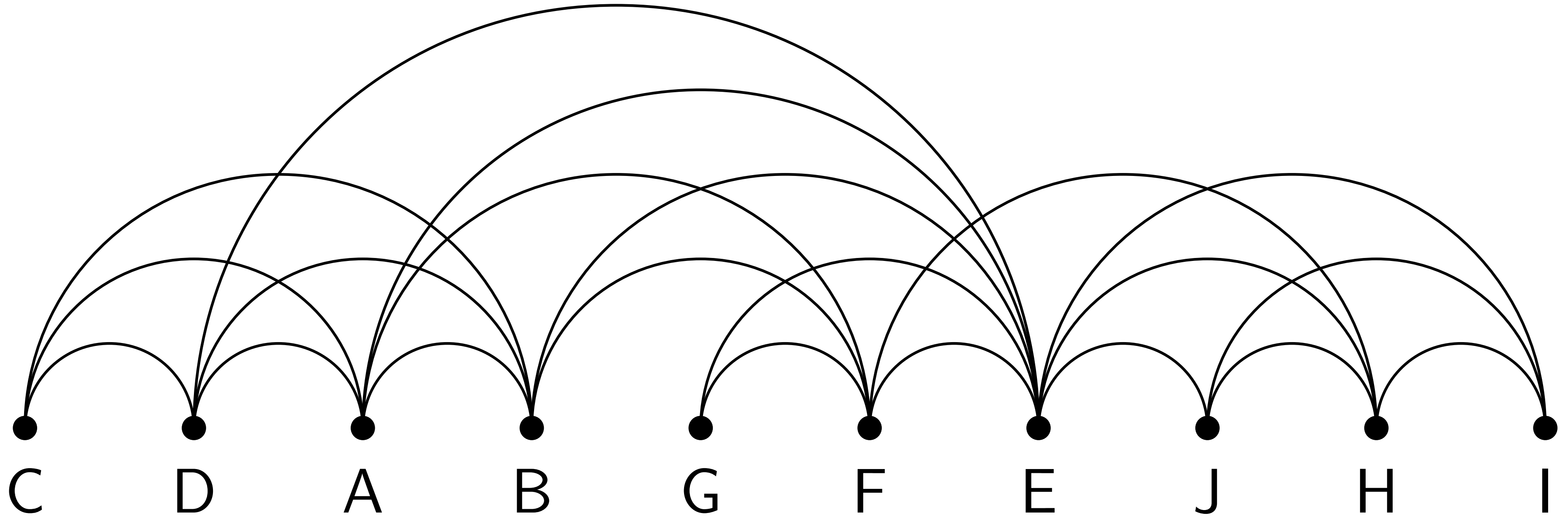
Input : lists $\mathrm{Adj}(v)$, $A(v)$.
Output : true, if $A(v) - \mathrm{Adj}(v) \neq \emptyset$, false otherwise.

```
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**



For **interval graphs** the **left-to-right ordering of right endpoints** in any interval representation gives a **PES**.



Input : chordal graph $G = (V, E)$.
Output : clique C and coloring ϕ.

compute with LexBFS a PES σ of G;
$C \leftarrow \emptyset$, $\phi \leftarrow 0$;
for $i \leftarrow n$ **to** 1 **do**
  $v \leftarrow \sigma(i)$;
  $X_v \leftarrow \mathrm{Adj}(v) \cap \{\sigma(i+1), \ldots, \sigma(n)\}$;
  $\phi(v) \leftarrow \min(\mathbb{N} - \{\phi(w) \mid w \in X_v\})$;
  if $|C| < |X_v + \{v\}|$ **then**
    $C \leftarrow X_v + \{v\}$;
  end if
end for
return C and ϕ;

Algorithm 5 : Compute $\omega(G)$ and $\chi(G)$

Input : chordal graph $G = (V, E)$.
Output : independent set U and clique cover ψ.

compute with LexBFS a PES σ of G;
$U \leftarrow \emptyset$, $\psi \leftarrow 0$;
for $i \leftarrow 1$ **to** n **do**
  $v \leftarrow \sigma(i)$, $X_v \leftarrow \mathrm{Adj}(v) \cap \{\sigma(i+1), \ldots, \sigma(n)\}$;
  if $\psi(v) = 0$ **then**
    $U \leftarrow U + \{v\}$;
    for $w \in X_v + \{v\}$ **do**
      $\psi(w) \leftarrow |U|$;
    end for
  end if
end for
return U and ψ;

Algorithm 6 : Compute $\alpha(G)$ and $\kappa(G)$