

```

procedure Parcours( $\mathcal{A}$ );
var X : sommet;
début
  X := racine( $\mathcal{A}$ );
  répéter
    si existe-fils(X) alors {
      empiler(X);
      X := premier-fils(X);
    }
    sinon {
      tant que  $\left\{ \begin{array}{l} X \neq \text{racine}(\mathcal{A}) \\ \text{et} \\ \text{non existe-frère}(X) \end{array} \right\}$  faire {
        X := dépiler();
      }
      si existe-frère(X) alors {
        X := frère(X);
      }
    }
  jusqu'à X := racine( $\mathcal{A}$ );
fin;

```

FIGURE 5.1 – Parcours en profondeur, itératif