

6.3.1 Parcours en profondeur itératif

```

procedure Parcours( $\mathcal{A}$ );
  var X : sommet;
  début
    [0]
    X := racine( $\mathcal{A}$ );
    [1]
  répéter
    si existe-fils(X) alors {
      [2]
      X := premier-fils(X);
      [3]
    }
    sinon {
      [4]
      tant que {
         $X \neq \text{racine}(\mathcal{A})$ 
        et
         $\text{non existe-frère}(X)$ 
      } faire {
        [5]
        X := père(X);
        [6]
      }
      si existe-frère(X) alors {
        [7]
        X := frère(X);
        [8]
      }
    }
  jusqu'à X := racine( $\mathcal{A}$ );
  [9]
fin;
    
```

6.3.2 Parcours en profondeur récursif

<pre> procedure parcour(X) ; begin [1] pour tout fils Y de X faire parcour(Y) ; [2] end </pre>	<pre> procedure parcour(X) ; begin [1] if (existe_fils(X)) parcour(premier_fils(X)) ; if (existe_frere(X)) parcour(frere(X)) ; [2] end </pre>
---	--