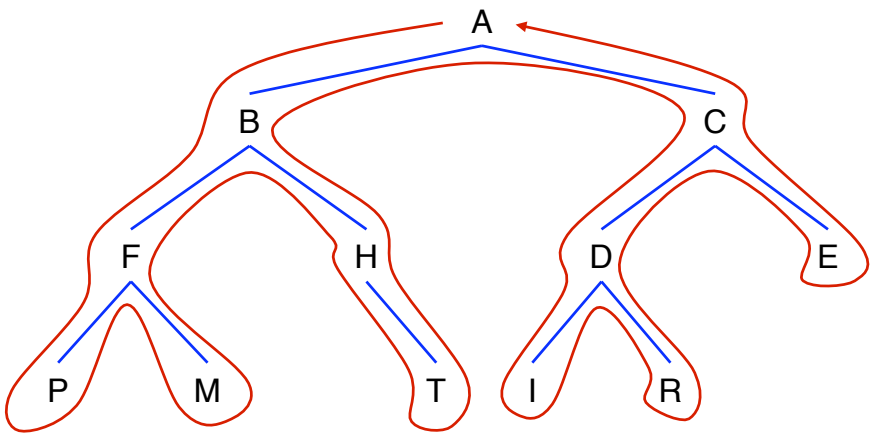
Algorithmique & programmation

Chapitre 5 : Arbres

Parcours

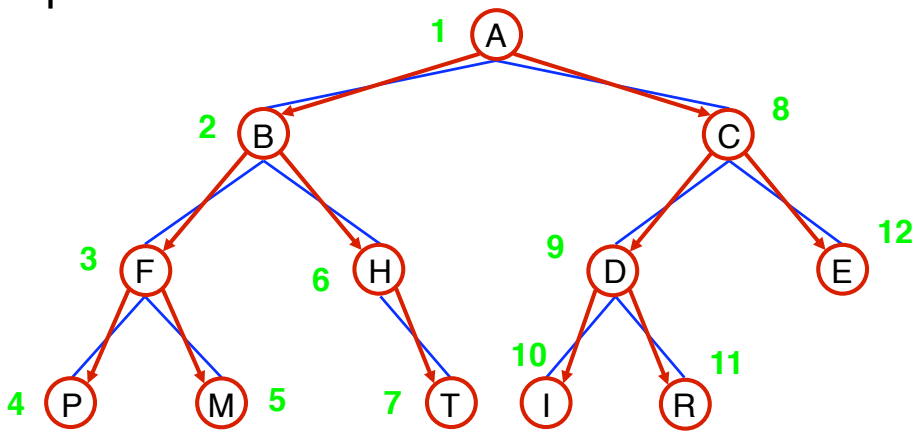
Parcours d'un arbre binaire

- Trois types de parcours
  - selon l'ordre d'examen de racine, fils gauche et fils droit



Parcours d'un arbre binaire

- Parcours préfixé (ou en préordre)
  - traiter la racine
  - parcourir le sous-arbre gauche
  - parcourir le sous-arbre droit



- Exemple : A B F P M H T C D I R E

Parcours préfixé d'un arbre binaire

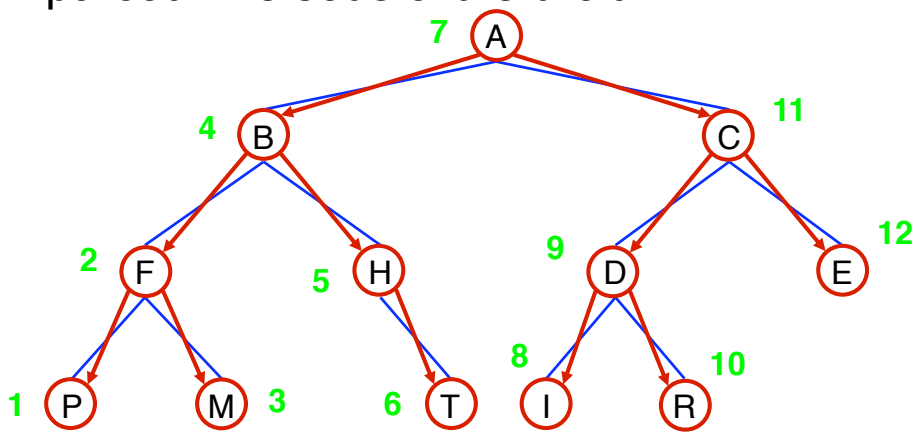
```
procédure préfixé (d racine : pointeur) ;
spécification { } ➔ {parcours préfixé de racine+}
debproc
    si racine ≠ nil alors
        traiter (racine) ;
        préfixé (racine↑.gauche) ;
        préfixé (racine↑.droite) ;
    finsi ;
finproc ;
```

Parcours préfixé d'un arbre binaire (**ada**)

```
procedure préfixé (racine : in Ta_Noeud) is
--spec { } ➔ {parcours préfixé de racine+}
begin
   if racine /= null then
     traiter (racine) ;
     préfixé (racine.all.gauche) ;
     préfixé (racine.all.droite) ;
   end if;
end préfixé ;
```


Parcours d'un arbre binaire

- Parcours infixé (ou projectif ou symétrique)
  - parcourir le sous-arbre gauche
  - traiter la racine
  - parcourir le sous-arbre droit



- Exemple : P F M B H T A I D R C E

Parcours infixé d'un arbre binaire

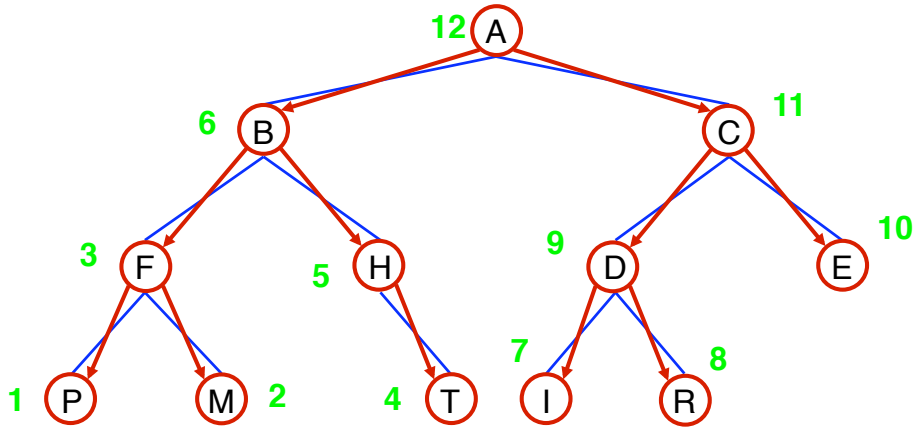
procédure infixé (d racine : pointeur) ;
spécification *{}➔{parcours infixé de racine⁺}*
debproc
    si racine ≠ nil alors
        infixé (racine↑.gauche) ;
        traiter (racine) ;
        infixé (racine↑.droite) ;
    finsi ;
finproc ;

Parcours infixé d'un arbre binaire (**ada**)

```
procedure infixé(racine : in Ta_Noeud) is
--spec { } ➔ {parcours préfixé de racine+}
begin
   if racine /= null then
     préfixé (racine.all.gauche) ;
     traiter (racine) ;
     préfixé (racine.all.droite) ;
   end if;
end infixé ;
```

Parcours d'un arbre binaire

- Parcours postfixé (ou terminal)
  - parcourir le sous-arbre gauche
  - parcourir le sous-arbre droit
  - traiter la racine



- Exemple : P M F T H B I R D E C A

Parcours postfixé d'un arbre binaire

procédure postfixé (d racine : pointeur) ;
spécification *{ }➔{parcours postfixé de racine⁺}*
debproc
  si racine ≠ nil alors
    postfixé (racine↑.gauche) ;
    postfixé (racine↑.droite) ;
    traiter (racine) ;
  finsi ;
finproc ;

Parcours postfixé d'un arbre binaire (**ada**)

```
procedure postfixé (racine : in Ta_Noeud) is
--spec { } ➔ {parcours préfixé de racine+}
begin
   if racine /= null then
     préfixé (racine.all.gauche) ;
     préfixé (racine.all.droite) ;
     traiter (racine) ;
   end if;
end postfixé ;
```