

Utilisation des éléments fictifs

Ajouter un élément fictif en tête de liste permet d'éviter d'avoir à traiter des cas particuliers pour l'insertion et la suppression d'éléments de la liste.

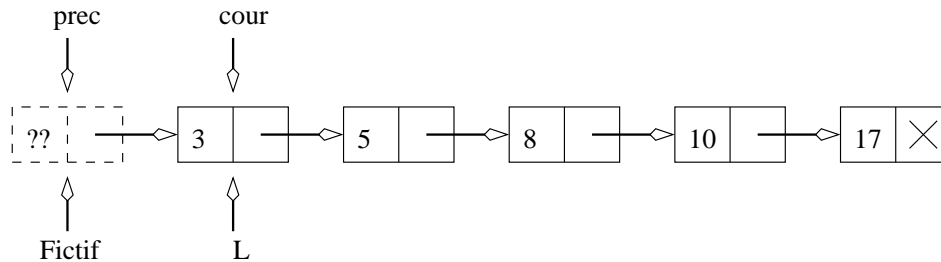


Figure 1: La liste avec un élément fictif ajouté en tête. `prec` et `cour` sont à leur position initiale, avant le parcours de la liste.

1 Insertion dans une liste

Supposons la liste triée dans l'ordre croissant. La procédure d'insertion s'écrit ainsi :

```
procedure Insertion(L : in out Liste; v : integer) is
  fictif, prec, cour, temp : Liste;
begin
  -- creation du fictif en tête de liste. Sa valeur n'a pas d'importance.
  fictif := new Elem'(0, L);

  -- initialisation. Invariant : prec.Suiv = cour
  prec := fictif;
  cour := L;

  -- parcours
  while cour /= null and then cour.Val < v loop
    prec := cour;
    cour := cour.Suiv;
  end loop;

  -- Ici, on sait qu'on doit intercaler v entre prec et cour, meme si cour est null
  temp := new Elem'(v, cour);
  prec.Suiv := temp;

  -- Attention : modifier L (si l'insertion s'est faite juste apres le fictif)
  L := fictif.Suiv;
end Insertion;
```

Avec un tel algorithme, l'insertion en tête de liste ou dans une liste vide n'est plus un cas particulier.

2 Suppression d'un élément

On suppose à nouveau la liste triée. On ne supprime ici que la première occurrence de v dans la liste.

```
procedure Suppression(L : in out Liste; v : integer) is
  fictif, prec, cour : Liste;
begin
  -- creation du fictif en tête de liste.
  fictif := new Elem'(0, L);

  -- initialisation. Invariant : prec.Suiv = cour
  prec := fictif;
  cour := L;

  -- parcours
  while cour /= null and then cour.Val < v loop
    prec := cour;
    cour := cour.Suiv;
  end loop;

  -- Ici, soit cour = null et on n'a pas trouvé v, soit cour.Val >= v
  if cour /= null and then cour.Val = v then
    -- On a trouve v, on l'enleve de la liste en faisant un "pont"
    prec.Suiv := cour.Suiv;
  end if;

  -- Attention: changer L (si la suppression s'est faite juste apres le fictif)
  L := fictif.Suiv;
end Suppression;
```

Ici aussi, la suppression du premier élément n'a pas à être gérée spécifiquement. La suppression dans une liste vide ne pose pas de problème non plus.

3 Remarque

Si l'on doit faire plusieurs insertions/suppressions, il peut être plus efficace de mettre une fois pour toute un fictif en tête de liste. Il faudra alors modifier les méthodes de parcours et d'affichage qui devront commencer *après* le fictif.