Validation d'algorithmes

Chapitre II

Preuve d'algorithmes : Exemples.

Raphaël Charrondière raphael.charrondiere@inria.fr

Ordre de Dershowitz–Manna dans un cas plus simple.

$$A >_{DM} B$$

- Rappel : un multi-ensemble est grosso-modo un ensemble pouvant contenir plusieurs fois le même élément.
- On considère l'ensemble E des multi-ensembles finis sur $\mathbb{N}$.

- Rappel : un multi-ensemble est grosso-modo un ensemble pouvant contenir plusieurs fois le même élément.
- On considère l'ensemble E des multi-ensembles finis sur $\mathbb{N}$.
- On défini $\phi : E \mapsto \mathbb{Z}$ qui à un multi-ensemble associe son élément maximum s'il est non vide ou -1 s'il est vide.

- Rappel : un multi-ensemble est grosso-modo un ensemble pouvant contenir plusieurs fois le même élément.
- On considère l'ensemble E des multi-ensembles finis sur $\mathbb{N}$.
- On défini $\phi : E \mapsto \mathbb{Z}$ qui à un multi-ensemble associe son élément maximum s'il est non vide ou -1 s'il est vide.
- Soit $A, B \in E$, on définit

$$A >_{DM} B \Leftrightarrow \phi(A \setminus (A \cap B)) > \phi(B \setminus (A \cap B))$$

- $\{3,2,1\} >_{DM} \{2,0\})$
- $\{4,4,2\} >_{DM} \{4,3,2\}$
- $\{2\} >_{DM} \{1,1,1,1,1,1,1\}$
- $\{5,5,3\} >_{DM} \{5,3\}$
- $A \neq \emptyset \implies A >_{DM} \emptyset$
- $B \subsetneq A \implies A >_{DM} B$

Ce que l'on doit montrer :

- $\forall A, B \in E,\ A = B \implies \neg(A >_{DM} B)$
- $\forall A, B \in E,\ A >_{DM} B \implies \neg(B >_{DM} A)$
- $\forall A, B, C \in E,\ (A >_{DM} B \wedge B >_{DM} C) \implies A >_{DM} C$
- Il n'existe pas de suite strictement décroissante infinie

À montrer : $\forall A, B \in E,\ A = B \implies \neg(A >_{DM} B)$.

Démonstration :

- Soit $A, B \in E$ avec $A = B$.
- On a $A \setminus (A \cap B) = B \setminus (A \cap B) = \emptyset$
- Donc $\phi(A \setminus (A \cap B)) = \phi(B \setminus (A \cap B)) = -1$
- Donc par définition de $>_{DM}$ on n'a pas $(A >_{DM} B)$

À montrer : $\forall A, B \in E,\ A >_{DM} B \implies \neg(B >_{DM} A)$

Démonstration :

- Soit $A, B \in E$, avec $A >_{DM} B$.
- Par définition de $>_{DM}$, $\phi(A \setminus (A \cap B)) > \phi(B \setminus (A \cap B))$.
- Donc on n'a pas $\phi(B \setminus (A \cap B)) < \phi(A \setminus (A \cap B))$
- Par définition de $>_{DM}$ on n'a donc pas $\neg(B >_{DM} A)$

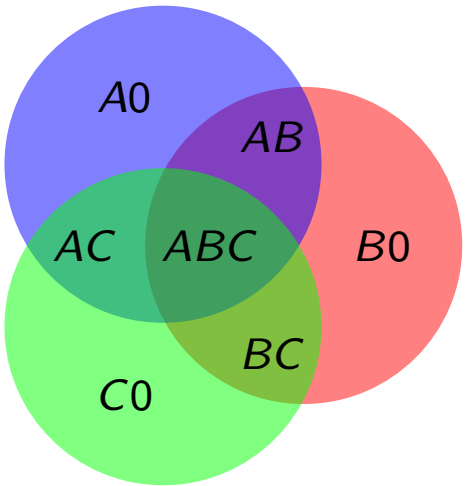
□

À montrer :

$\forall A, B, C \in E,\ (A >_{DM} B \wedge B >_{DM} C) \implies A >_{DM} C$

Démonstration :

- Soit $\forall A, B, C \in E$ avec $A >_{DM} B \wedge B >_{DM} C$



- $A = A0 \cup AB \cup AC \cup ABC$, etc.
- Par définition de $>_{DM}$ on peut écrire
- (1) $\forall x \in B0 \cup BC,\ \exists y \in A0 \cup AC,\ y > x$
- (2) $\forall z \in C0 \cup AC,\ \exists w \in AB \cup B0,\ w > z$

Prouvons (3) $\forall t \in AC,\ \exists x \in A0 \cup AB,\ x > t$

- Supposons $AC \neq \emptyset$
- Prenons $e = \max AC$, (2) $\Longrightarrow\ \exists w \in AB \cup B0,\ w > e$.
- Cas $w \in B0$, (1) $\Longrightarrow\ \exists z \in A0 \cup AC,\ z > w$, par transitivité $z > e$, comme e est le max de AC $z \in A0$.
- Donc $\exists x \in A0 \cup AB,\ x > e$.
- Dans le cas où $AC = \emptyset$ la proposition est trivialement vraie.

Soit $x \in C0 \cup BC$

Cas 1 : $x \in C0$

- $(2) \Longrightarrow \exists t \in AB \cup B0,\ t > x$
- Cas $t \in B0$, $(1) \Longrightarrow \exists z \in A0 \cup AC,\ z > t$
- Sous-cas $z \in AC$, $(3) \Longrightarrow \exists w \in A0 \cup AB,\ w > z$
- Donc $\exists r \in A0 \cup AB,\ r > x$.

Cas 2 : $x \in BC$ Similaire...

Soit $x \in C0 \cup BC$

Cas 1 : $x \in C0$

- $(2) \Longrightarrow \exists t \in AB \cup B0,\ t > x$
- Cas $t \in B0$, $(1) \Longrightarrow \exists z \in A0 \cup AC,\ z > t$
- Sous-cas $z \in AC$, $(3) \Longrightarrow \exists w \in A0 \cup AB,\ w > z$
- Donc $\exists r \in A0 \cup AB,\ r > x$.

Cas 2 : $x \in BC$ Similaire...

Conclusion :

- $\forall x \in C0 \cup BC,\ \exists y \in A0 \cup AB,\ y > x$
- c'est-à-dire $\phi(A0 \cup AB) > \phi(C0 \cup BC)$
- soit $\phi(A \setminus (A \cap C)) > \phi(C \setminus (A \cap C))$
- on a bien $A >_{DM} C$

À montrer : Il n'existe pas de suite strictement décroissante infinie

Démonstration : On va le montrer par récurrence. Posons la propriété $P(n)$: « il n'y a pas de suite strictement décroissante infinie dont le premier terme A est tel que $\phi(A) \leqslant n$ »

- Initialisation $n = 0$: Soit A, tel que $\phi(A) \leqslant 0$. On distingue deux cas :

À montrer : Il n'existe pas de suite strictement décroissante infinie

Démonstration : On va le montrer par récurrence. Posons la propriété $P(n)$: « il n'y a pas de suite strictement décroissante infinie dont le premier terme A est tel que $\phi(A) \leqslant n$ »

- Initialisation $n = 0$: Soit A, tel que $\phi(A) \leqslant 0$. On distingue deux cas :
  - $A = \emptyset$: il n'y a pas d'éléments strictement inférieur à $\emptyset$ pour $>_{DM}$

À montrer : Il n'existe pas de suite strictement décroissante infinie

Démonstration : On va le montrer par récurrence. Posons la propriété $P(n)$: « il n'y a pas de suite strictement décroissante infinie dont le premier terme A est tel que $\phi(A) \leqslant n$ »

- Initialisation $n = 0$: Soit A, tel que $\phi(A) \leqslant 0$. On distingue deux cas :
  - $A = \emptyset$: il n'y a pas d'éléments strictement inférieur à $\emptyset$ pour $>_{DM}$
  - $A = \{\text{"k fois l'élément 0"}\}$: il y a seulement k éléments de E inférieurs à A qui sont "j fois l'élément 0" où $0 \leqslant j < k$

  Donc il n'y a pas de suite strictement décroissante infinie commençant par A.

Supposons $P(n-1)$ et montrons $P(n)$: Soit A, tel que $\phi(A) \leqslant n$

- $\phi(A) < n$: d'après $P(n-1)$ il n'y a pas de suite strictement décroissante infinie commençant par A.

Supposons $P(n-1)$ et montrons $P(n)$: Soit A, tel que $\phi(A) \leqslant n$

- $\phi(A) < n$: d'après $P(n-1)$ il n'y a pas de suite strictement décroissante infinie commençant par A.
- $\phi(A) = n$: Soit une suite strictement décroissante commençant par A.
  - Si elle contient un élément A_i, tel que $\phi(A_i) < n$, par $P(n-1)$ cette suite est finie.

Supposons $P(n-1)$ et montrons $P(n)$: Soit A, tel que $\phi(A) \leqslant n$

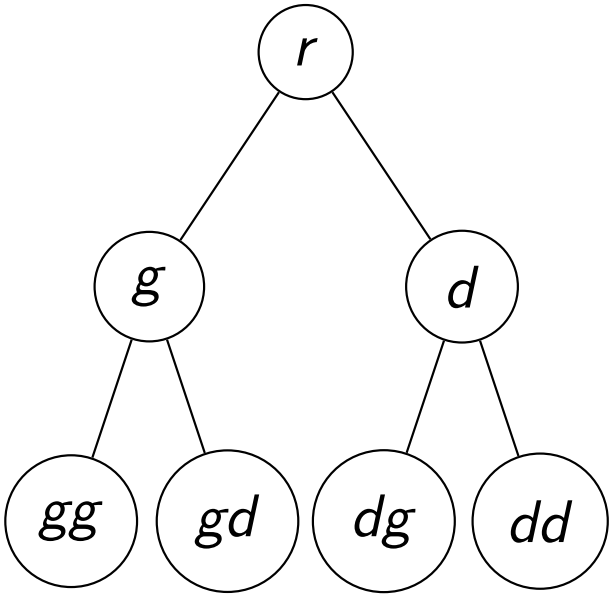
- $\phi(A) < n$: d'après $P(n-1)$ il n'y a pas de suite strictement décroissante infinie commençant par A.
- $\phi(A) = n$: Soit une suite strictement décroissante commençant par A.
  - Si elle contient un élément A_i, tel que $\phi(A_i) < n$, par $P(n-1)$ cette suite est finie.
  - Sinon chaque terme de la suite possède l'élément n. Par définition de $>_{DM}$ on peut supprimer cet élément à chaque terme sans modifier la décroissance stricte de la suite. On répète cette suppression jusqu'à ce qu'un terme de la suite ne contienne plus de n. La nouvelle suite obtenue est toujours strictement décroissante et elle est de même longueur. D'après le cas précédent cette nouvelle suite est finie, donc la suite d'origine aussi.

Donc il n'y a pas de suite strictement décroissante infinie commençant par A.

- Remplacer $\mathbb{N}$ par tout ordre bien fondé. (Il faut adapter la définition de $>_{DM}$ ou changer ϕ. . .)
- Est-ce que $>_{DM}$ est total ?
- Lire la vrai définition et faire le lien avec ce cas.

Arbre binaire de recherche

Définition : arbre binaire



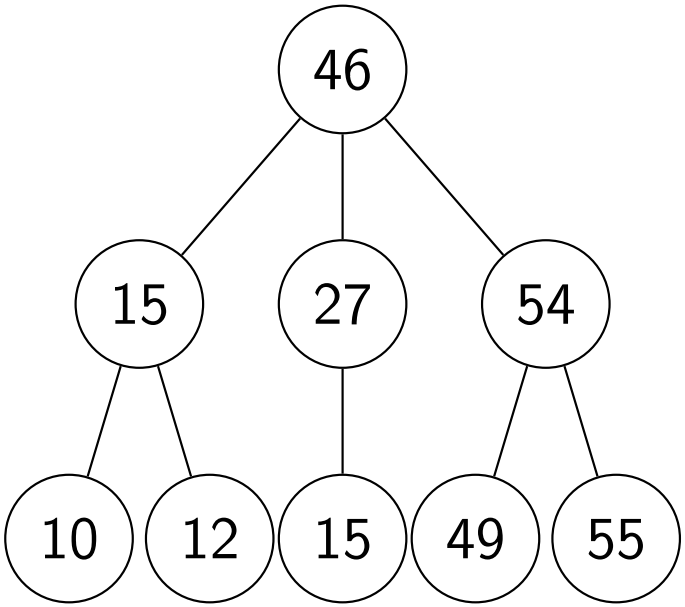
- Chaque nœud a optionnellement un fils gauche et/ou un fils droit.
- Pour chaque nœud n on définit $\mathcal{G}(n)$ l'ensemble contenant son fils gauche ng et des héritiers du fils gauche, cet ensemble est vide si le fils gauche n'existe pas
- De même $\mathcal{D}(n)$ à droite

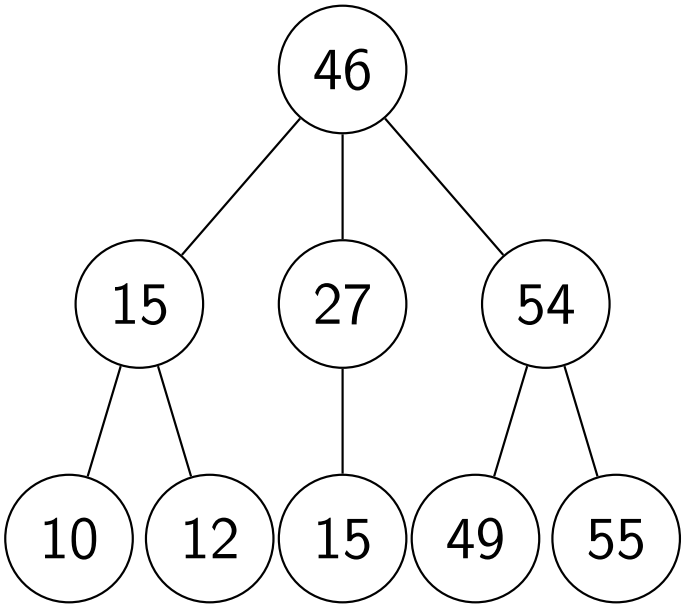
Soit un ensemble E muni d'une relation d'ordre $<$ totalement ordonné.

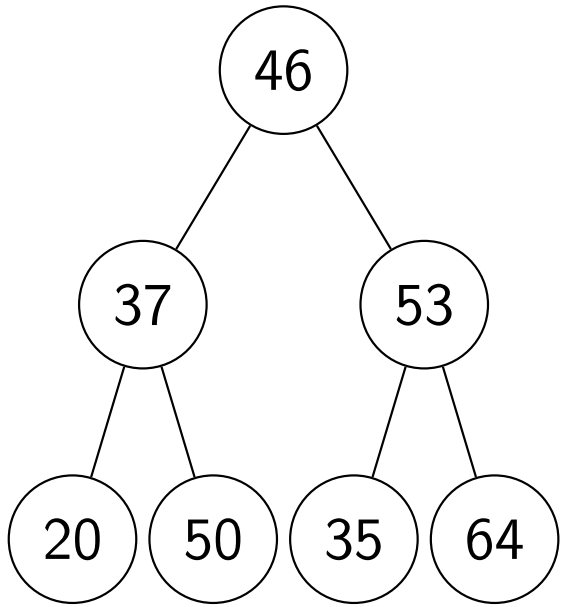
Remarque : *On va assimiler les nœuds aux éléments qu'ils contiennent, c'est à dire que si le nœud n_1 contient l'élément e_1, et n_2 contient e_2, alors $n_1 < n_2 \Leftrightarrow e_1 < e_2$*

Arbre binaire de recherche

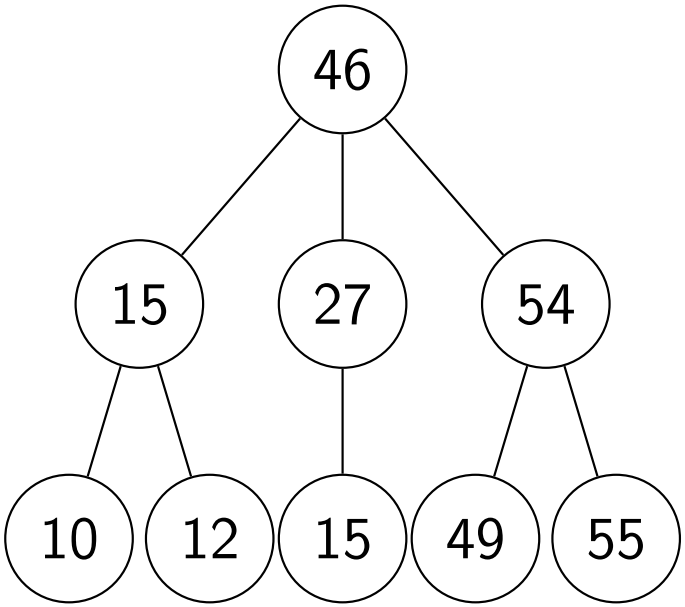
- Est un arbre binaire
- Pour tout nœud n $\forall g \in \mathcal{G}(n) g < n$
- Pour tout nœud n $\forall d \in \mathcal{D}(n) n < d$


46
15
27
54
10
12
15
49
55

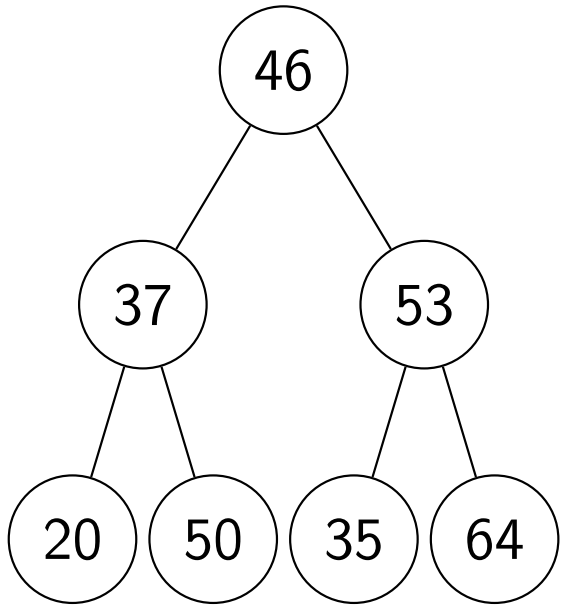




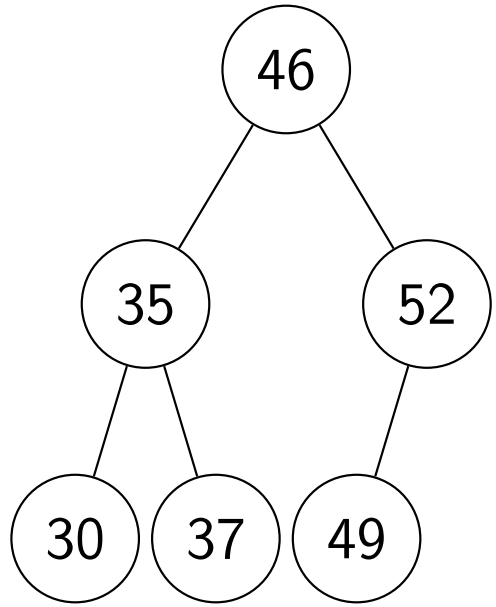
Pas binaire

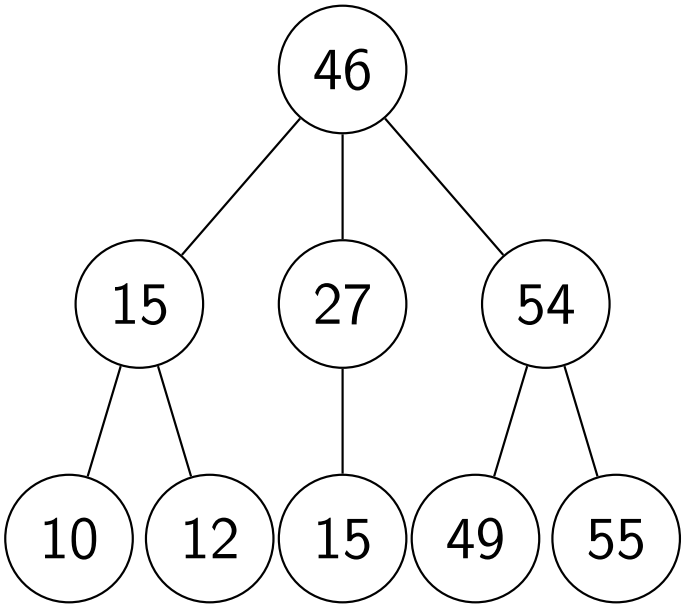


Pas binaire

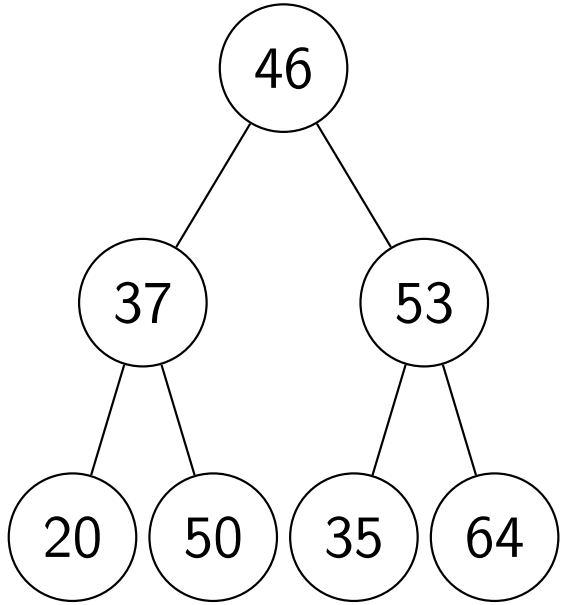


Pas un arbre de recherche

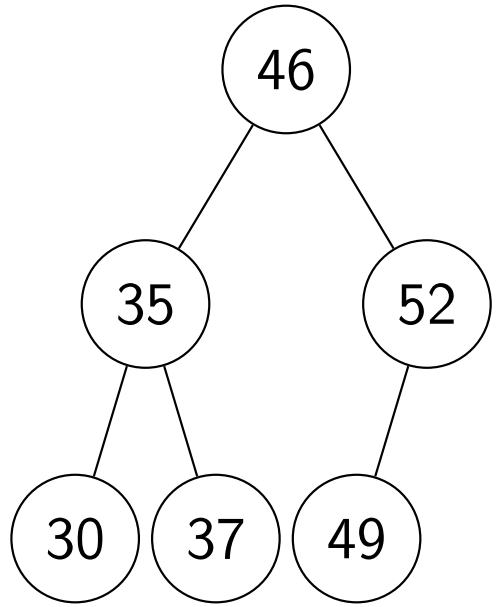




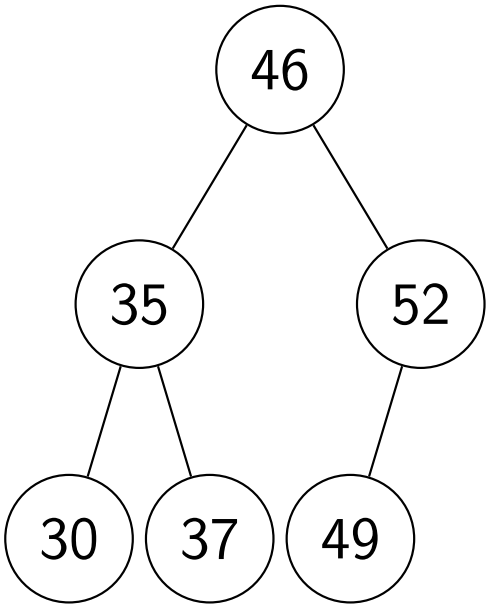
Pas binaire



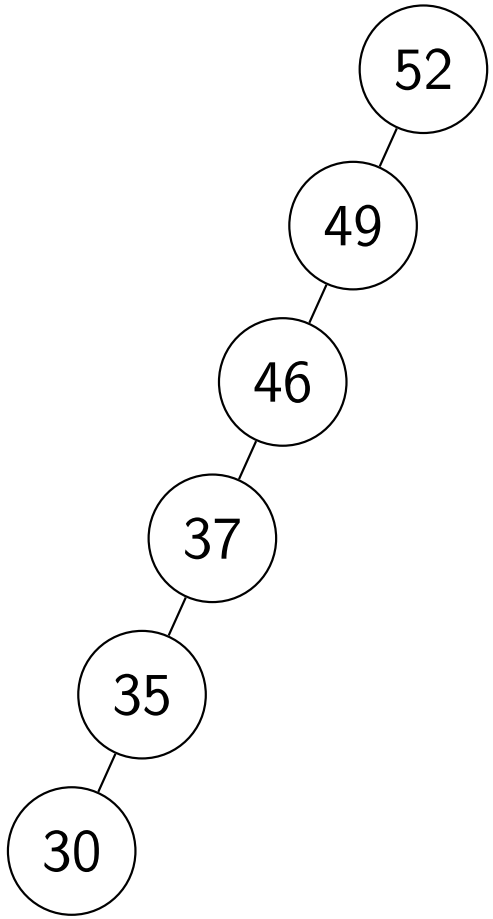
Pas un arbre de recherche



ABR



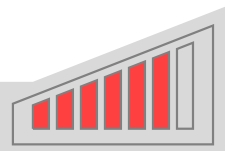
ABR précédent



ABR profond

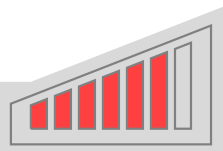
- Les deux arbres ont les mêmes éléments, et sont des ABR
- L'un est plus profond
- Si n est le nombre d'élément la profondeur de l'ABR est environ entre n et $\log_2 n$

```
void insert(Tree* t, Element e) {
  Node* root = t->root;
  while (root) {
    Node* next;//Navigate in tree
    if(root->value<e){
      next = root->left;
      if(!next) root->CreateLeft(e);
    }
    else if(root->value>e){
      next = root->right;
      if(!next) root->CreateRight(e);
    }
    else//e is already in tree
      root=nullptr;
    root=next;
  }
}
```

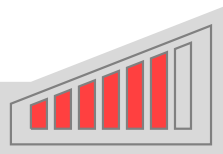



Et maintenant ?

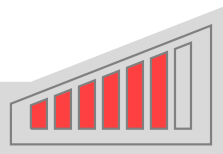
- **Variant**
- **Invariant**



- **Variant** ϕ : «Le nombre d'héritier de `root`»
- **Invariant**



- **Variant** ϕ : «Le nombre d'héritier de `root`»
- **Invariant** Plus complexe. On veut traduire
  - L'ABR ne change pas, sauf pour e qui peut être inséré
  - D'ailleurs e ne change pas non plus
  - L'élément sera inséré parmi les héritiers de `root`
  - L'insertion de e ne perturbe pas les propriétés de l'ABR.



- **Variant** ϕ : «Le nombre d'héritier de `root`»
- **Invariant** Plus complexe. On veut traduire
  - L'ABR ne change pas, sauf pour e qui peut être inséré
  - D'ailleurs e ne change pas non plus
  - L'élément sera inséré parmi les héritiers de `root`
  - L'insertion de e ne perturbe pas les propriétés de l'ABR.

Remarque On pourrait se contenter de dire que le nouvel arbre contient les même éléments et peut-être e, et que c'est un ABR, malheureusement nous avons vu que la profondeur d'un ABR peut varier, il peut être judicieux de garder une propriété sur la taille de l'ABR. Ici, la taille grandira d'au plus 1

Rappel de notation : $m_0(a) = a_0$
Rappel : un invariant est une propriété $P(m_0, m)$. m_0 est l'état mémoire au début du bloc d'instruction considéré (ici, la boucle `while`) et on suppose $P(m_0, m_0)$
$P(m_0, m)$:

- $e = e_0$
- $e \in t_0 \implies t = t_0$
- $e \notin t_0 \implies t \setminus \{e\} = t_0$
- $\forall \text{Node } n \in t,\ e \in \mathcal{G}(n) \implies e < n$
- $\forall \text{Node } n \in t,\ e \in \mathcal{D}(n) \implies n < e$

Ce niveau de formalisme vous est demandé