

Arbres binaires de recherche et AVL

Alexandre Meyer

Motivation

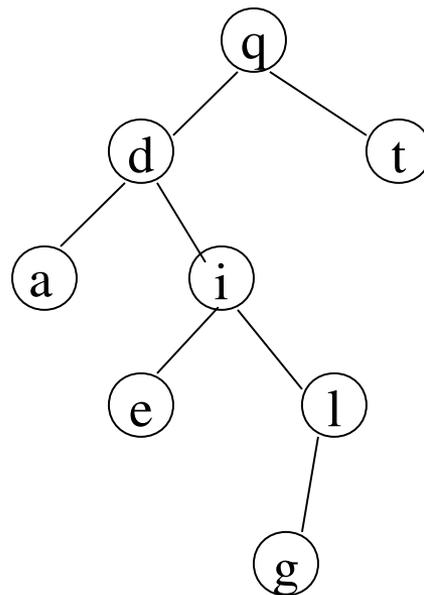
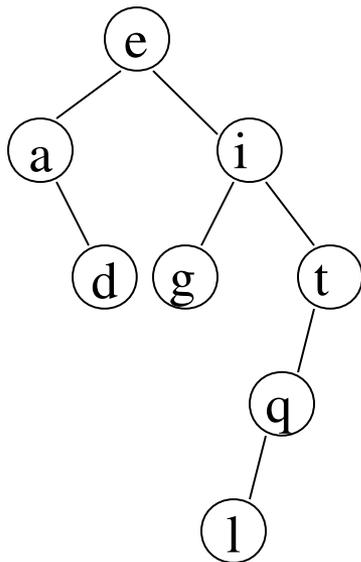


- Recherche dichotomique ou par interpolation dans un tableau trié
 - solution satisfaisante
 - $O(\log n)$

- Insertion et suppression dans un tableau trié
 - non satisfaisante
 - $O(n)$

ABR : Définitions

- Un **arbre binaire de recherche** est un arbre étiqueté tel que pour tout nœud v de l'arbre
 - les éléments du sous-arbre gauche de v sont inférieurs à v
 - les éléments du sous-arbre droit de v sont supérieurs à v



le parcours symétrique
→ la suite des éléments
triée en ordre croissant

TDA ABR + implementation

TDA abr (Arbres Binaires de Recherche)

utilise element (doit définir un ordre), bool

Opérations

abrVide → arbrebin

abrEstVide(a : abr) → bool

abrRecherche(A : abr; x : element) → *element

abrInsertFeuille(A : donnée-résultat abr; x : element) → abr

abrMin(a : abr) → *element

Type noeudabr = **record**

data : *element

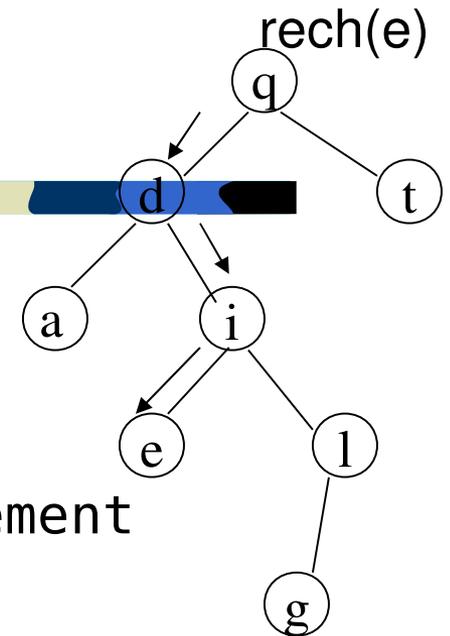
gauche : *noeudabr

droit : *noeudabr

Type abr = *noeudabr

Les fonctions internes à ce TDA
sont celles du TDA arbre binaire

Recherche récursive



```
abrRechercher(A : abr; x : element) → *element
  si A==abrVide alors → null
  sinon
    si *abrDonnee(A)>x
      alors →abrRechercher( abrGauche(A), x)
    sinon
      si *abrDonnee(A)<x
        alors →abrRechercher( abrDroit(A), x)
      sinon
        → &(A->data)
```

Recherche itérative

```
abrRechercher_iter(A : abr; x : elt) → *element
```

```
  p ← A
```

```
  si A==abrVide alors → null
```

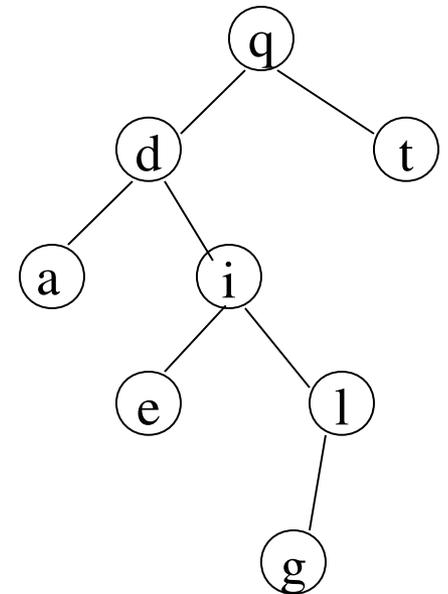
```
  tant que p != null et x != *abrDonnee(p) faire  
    si x < abrDonnee(p)
```

```
      alors p ← abrGauche(p)  
      sinon p ← abrDroit(p)
```

```
  ftq
```

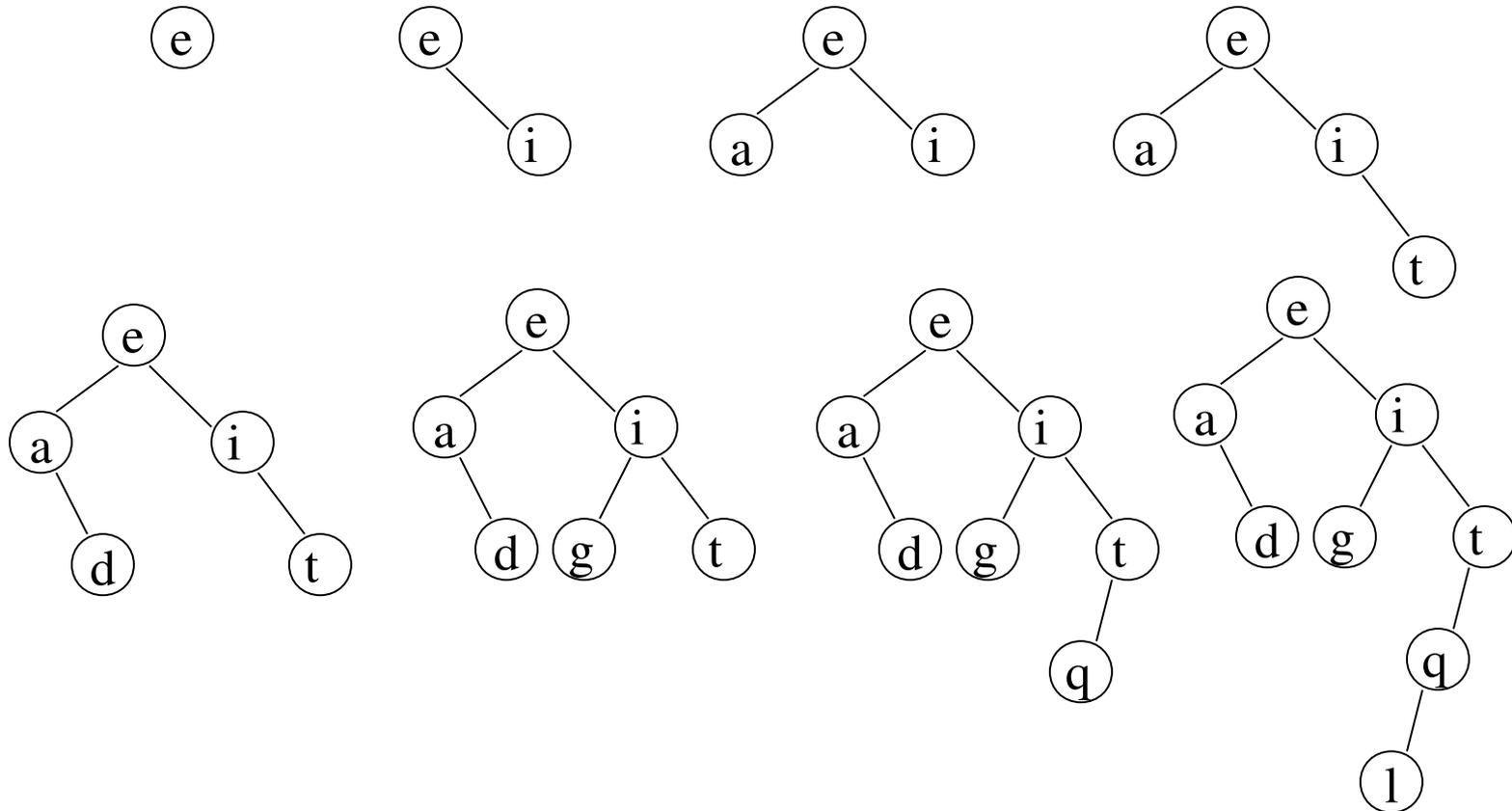
```
  si p==null alors → null
```

```
  sinon →&(p->data)
```



Insertion aux feuilles

- Descente dans l'arbre comme pour la recherche mais jusqu'en bas
- Ajout de l'élément à insérer



Insertion aux feuilles : algo



`abrInsertFeuille(A : donnée-résultat abr; x : *element) → abr`

- Possible récursivement
- Mais aussi simple itérativement

Insertion aux feuilles : algo

abrInsertFeuille(A : donnée-résultat abr; x : element) → abr

p , perep, feuille : abr

perep ← null

p ← A

tant que p != null **faire**

 perep ← p

si x < abrDonnee(p)

alors p ← abrGauche(p)

sinon p ← abrDroit(p)

ftq

feuille ← abrCons(x, abrVide, abrVide)

si perep == null **alors** →feuille

si x < perep->data

alors *perep.gauche ← feuille **sinon** *perep.droit ← feuille

→A