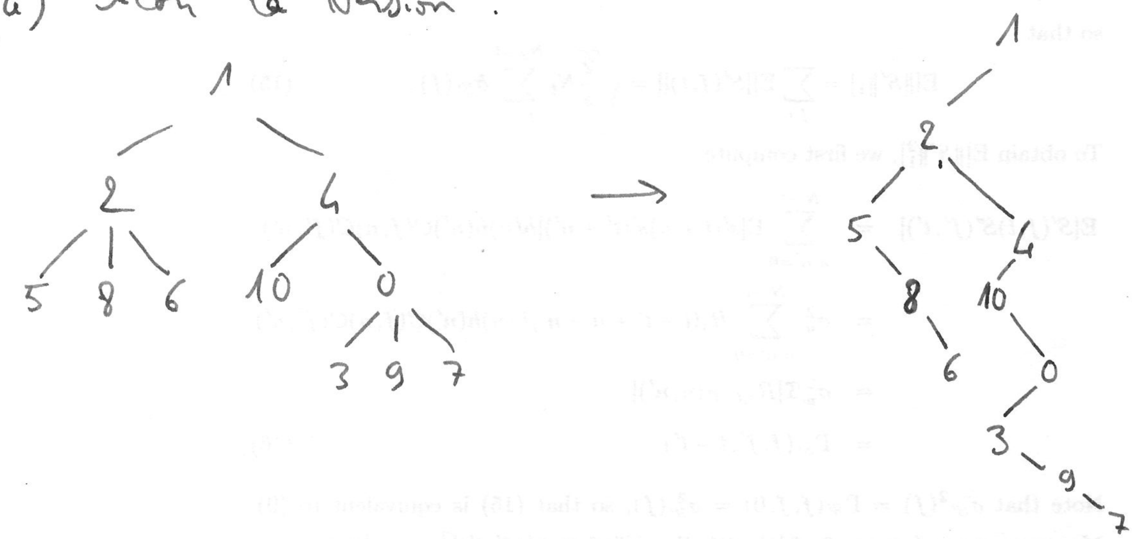
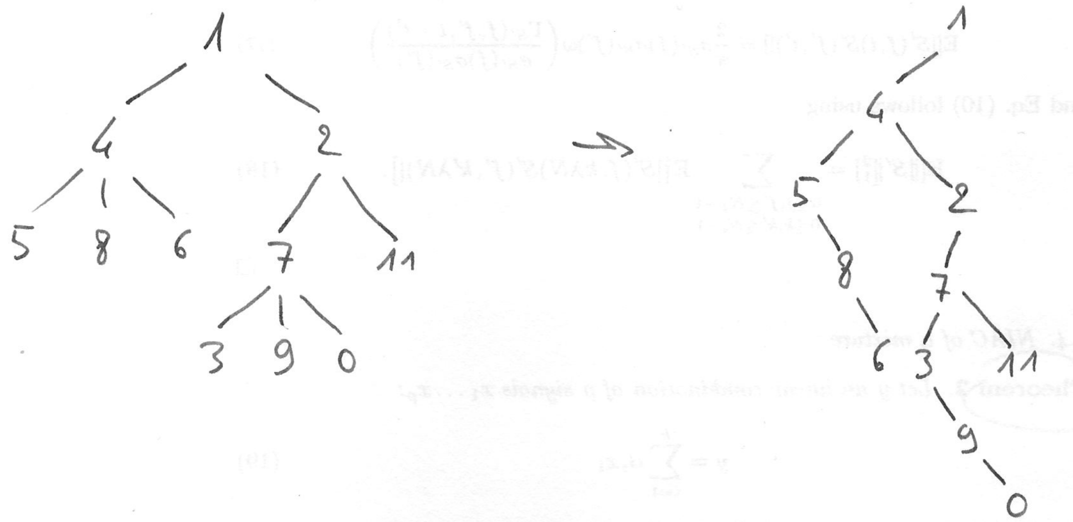
Correction de l'examen du 8.1.2020

1) Questions de cours : voir cours

2) Représentation d'arbres

a) Selon la version :

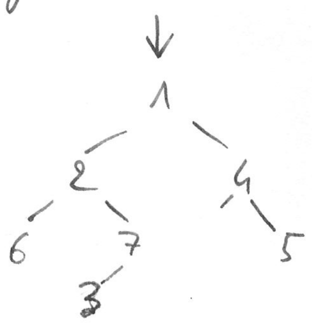




b) Selon la version :

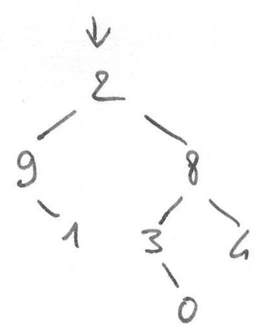
Pr. : 1267345

Sy : 6237145



Pr : 2918304

Sy : 9123084



③ Itératif → récursif

Algorithme rechDichoDernier

```
DEBUT
  /* Entrées : un vecteur V, un élément x, des bornes inf et sup */
  /* Sortie : i si x apparaît au rang i dans V, 0 sinon */
  Si inf = sup,
     Si V(inf) = x, retourne inf
     Sinon, retourne 0
  Sinon
     med ← (inf + sup + 1) div 2
     Si x < V(med)
        retourne rechDichoDernier(V, x, inf, med-1)
     Sinon
        retourne rechDichoDernier(V, x, med, sup)
FIN
```

(4) Démonstration d'une propriété des ABR

Posons $(E, \leq)$ l'ensemble des arbres binaires muni de la relation d'ordre "est un sous-arbre de"

1- $\forall E' \subset E$, tout arbre A de E' ayant le plus petit nombre de nœuds est minimal, au sens où aucun autre arbre de E' ne peut être sous-arbre de A

Tout sous-ensemble de E admettant un élément minimal, $(E, \leq)$ est bien fondé.

2- Pour un arbre binaire A, posons :

$P(A)$ = " Si l'algo affiche les nœuds de A en ordre croissant, alors A est un ABR "

$P(\emptyset)$ est vrai

3- Soit $A \in E$, tel que tous ses nœuds ont des étiquettes distinctes

Supposons que $\forall B < A \quad P(B) \quad (*)$

Supposons que l'algo affiche les nœuds de A en ordre croissant

Il affiche :

$\underbrace{\text{parcoursSym}(G(A))}_{G(A) \text{ en ordre croissant}} - \text{racine}(A) - \underbrace{\text{parcoursSym}(D(A))}_{D(A) \text{ en ordre croissant}}$

Comme $G(A)$ et $D(A) < A$, $P(G(A))$ et $P(D(A))$ (d'après hypothèse $(*)$)

Donc $G(A)$ et $D(A)$ sont des ABR

Comme l'affichage de A est en ordre croissant,

racine(A) est $>$ aux nœuds de $G(A)$
$<$ ——— $D(A)$

Cette propriété + le fait que $G(A)$ et $D(A)$ sont ABR $\Rightarrow$ A est ABR

4- On a montré que

$\forall A \in E, (\forall B < A, P(B)) \Rightarrow P(A)$

Donc $\forall A \in E$ (avec des étiquettes distinctes), $P(A)$

(5) Complexité d'un algorithme récursif

a) Pour $n > 1$, l'algo enveloppe effectue :

- αn comparaisons dans divise (P)
- $C\left(\left\lfloor \frac{n}{2} \right\rfloor\right)$ c. dans l'appel récursif enveloppe (P_1)
- $C\left(\left\lceil \frac{n}{2} \right\rceil\right)$ comparaisons dans l'appel récursif enveloppe (P_2)
- βn comparaisons dans fusion (E_1, E_2)

Donc $$C(n) = C\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + C\left(\left\lceil \frac{n}{2} \right\rceil\right) + (\alpha+\beta)n$$

b) Approchons n par une puissance de 2 : $n = 2^p$ et posons $\gamma = \alpha + \beta$. On a alors :

$$C(2^p) = 2\,C(2^{p-1}) + \gamma 2^p$$

En posant $x(p) = C(2^p)$,

$$x(p) = 2x(p-1) + \gamma 2^p$$

Soit $X(z)$ la série génératrice associée à x

$$X(z) = 2z\,X(z) + \gamma \sum_{p\geq 0} 2^p z^p$$

$$(1-2z)\,X(z) = \frac{\gamma}{1-2z}$$

$$X(z) = \frac{\gamma}{(1-2z)^2}$$

Donc $$x(p) = \gamma(p+1)2^p$$

$$C(n) = \gamma(\log_2 n + 1)n \sim \gamma n \log_2 n$$