Algorithmique et Structures de données 1

LIV 2022-2023

Travaux Dirigés 6

Site du cours : `https://defelice.up8.site/algo-struct.html`
Les exercices marqués de (@) sont à faire dans un second temps.

On utilise la structure suivante pour représenter les arbres.

```
typedef struct s_noeud_t
{
   int v; // étiquette du noeud (v pour valeur)
   struct s_noeud_t* g; // pointeur vers la racine du sous-arbre  gauche
   struct s_noeud_t* d; // pointeur vers la racine du sous-arbre droit
} noeud_t;

// l'arbre vide aura la valeur NULL
```

Exercice 1. *Comment faire pousser un arbre ?*

La fonction `noeud_t* consA(noeud_t* g,noeud_t* d,int v)` construit un nouvel arbre en assemblant deux arbres `g` `d` par un noeud d'étiquette `v`. Dessinez l'arbre construit et renvoyé par l'appel suivant :

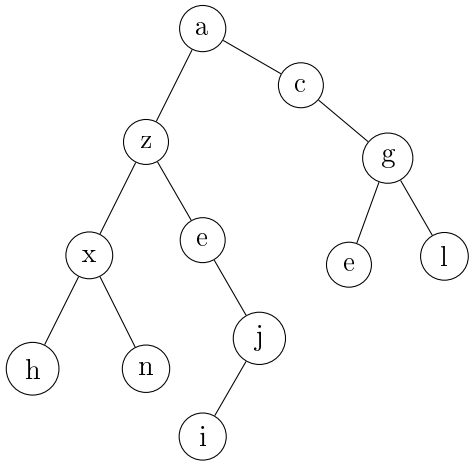
`consA(consA(NULL,NULL,4),consA(consA(NULL,NULL,1),NULL,7),2)`

Exercice 2. *Taille*

Écrire une fonction `int taille(noeud_t* arbre)` qui renvoie le nombre de noeuds d'un arbre.

Exercice 3. *Parcours d'un arbre*

Donner la liste des nœuds de l'arbre binaire étiqueté suivant avec un parcours suffixe (ou postfixé) (gauche,droit,racine).



Les arbres vides ne sont pas représentés sur la figure.

Exercice 4. *Reconstruire*

On a parcouru un arbre en préfixe et on a obtenu la liste suivante :

`a b d` $\perp$ $\perp$ `c e` $\perp$ `f h` $\perp$ $\perp$ $\perp$ `g` $\perp$ $\perp$ $\perp$

Reconstruire et dessiner l'arbre associé.