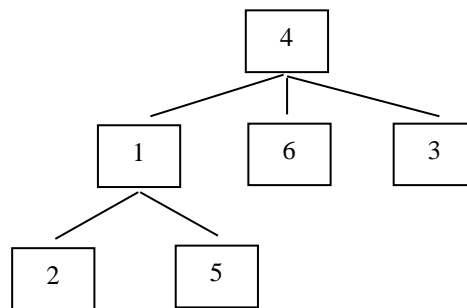


Travaux Dirigés Séance nr. 6

Implantation d'une forêt générale par liste chaînée de fils

On souhaite représenter des arbres quelconques : chaque noeud peut avoir n fils, n n'étant pas borné. Pour cela on utilise la binarisation d'un arbre tel que vu en cours : concrètement, en chaque nœud on stocke un pointeur vers le premier fils et un pointeur vers le frère (soit deux pointeurs seulement, comme pour un arbre binaire !).

1. Faire une représentation « binarisée » de l'arbre suivant :



2. On se propose d'utiliser la structure de données suivante:

```

typedef struct s_noeud {
    T      val;
    struct s_noeud  *fils, *frere;
} noeud, *arbre;
  
```

Proposer une implantation des opérations de base :

```

arbre      arbrevide() ;
arbre      nouvnoeud(T x, arbre frere, arbre fils); // création
arbre      fils(arbre a); // récupérer le premier fils
arbre      frere(arbre a) ; // récupérer le frère
bool       est_vide(arbre a)
bool       est_feuille(arbre a) ;
bool       est_fils(T x, arbre a) ; // teste si x est un des fils
                                              // de a (fils, pas descendant)
unsigned int nb_fils(arbre a); // nombre de fils de a,
                               // dans l'exemple « 4 » a trois fils
  
```

En utilisant ces opérations faire également un « main » qui construit l'arbre de la question 1.

3. Ajouter maintenant des opérations pour :

- calculer la hauteur de l'arbre (attention ce n'est pas la hauteur de l'arbre binaire). La hauteur de l'arbre de l'exemple est 3.
- tester l'existence d'une étiquette. Par exemple 5 existe dans l'arbre précédent : on renvoie alors le pointeur vers le nœud qui contient 5 (un pointeur nul si le nœud n'existe pas).
- écrire un algorithme permettant d'afficher l'arbre de la façon suivante :

```

/4
  /1
    /2
      /5      ...
        /6
          /3

```

A chaque niveau de profondeur on décale l'affichage vers la droite. Pour chaque nœud on affiche ses fils les uns en dessous des autres. On fait également précéder d'un caractère pipe chaque affichage.