

# Rapport TP6

---

Je n'ai pas pu faire le dernier exercice en raison d'un manque de temps.

## Choix d'implémentation

---

### Fonction supplémentaires :

- Pour des soucis de lisibilité j'ai décidé de rajouter des fonctions comme par exemple :
  - isRightSon
  - isLeftSon
  - redSiblings
  - redParentAndUncle
  - redRoot
  - redAndParentToo

### Fonction bstree\_searchv2 :

- fonction qui renvoie une struct Couple afin de faire uniquement un seul search et pouvoir obtenir la position de la valeur en plus de savoir si elle a été trouvée (le but étant de réduire la complexité)

```
typedef struct {  
    bool found;  
    ptrBinarySearchTree pos;  
} Couple;
```

### Fonction bstree\_remove\_nodev2 :

- Fonction faisant appel à remove\_node afin d'être compatible avec le format de OperateFuncion

### Fonction printNode :

- le type NodeColor ainsi que la fonction printNode sont déclarés dans bstree.c car l'utilisateur n'a en aucun cas besoin de l'utiliser.
- création d'une chaîne de caractère color car une enum renvoie un int et non pas le nom de l'élément.

```
char *color = t->color ? "red" : "black";
```

- texte de couleur blanche afin d'être visible dans les nodes noirs.

### Fonction `rbtree_export_do` :

- nodes de couleurs blanches ( même raison que pour le texte )

### Fonction `rightrotate` :

- Cette fonction est déclarée uniquement dans `bstree.c` car l'utilisateur n'a en aucun cas besoin de l'utiliser. En l'utilisant il risque uniquement de déséquilibrer l'arbre.

### Description du comportement sur les jeux de tests fournis

- Tous les jeux de test ont été validés avec les 3 fichiers

### Analyse personnelle du travail effectué

- Il n'y a aucune fuite de mémoire et à ma connaissance aucune erreur.
- Pour l'équilibrage de l'arbre mon algorithme fonctionne mais n'est pas idéal. On voit que pour `testsimple1` il y a 4 rouges alors qu'idéalement on ne devrait en obtenir que 3 rouges comme sur l'exemple suivant : [Si le lien n'affiche pas l'arbre correctement veuillez copier coller le code suivant](#)

```
digraph RedBlackTree {
  graph[ ranksep =0.5];
  node [ shape = record, color = white];

  n1 [style=filled, fillcolor=red, fontcolor = white, label="{{<parent>}}1|{{<left>|<right>}}"];
  ln1l1 [style=filled, fillcolor=grey, label="NIL"];
  n1:left:c -> ln1l1:n [headclip=false, tailclip=false]
  rn1l1 [style=filled, fillcolor=grey, label="NIL"];
  n1:right:c -> rn1l1:n [headclip=false, tailclip=false]
  n2 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}2|{{<left>|<right>}}"];
  n2:left:c -> n1:parent:c [headclip=false, tailclip=false]
  rn1l2 [style=filled, fillcolor=grey, label="NIL"];
  n2:right:c -> rn1l2:n [headclip=false, tailclip=false]
  n3 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}3|{{<left>|<right>}}"];
  n3:left:c -> n2:parent:c [headclip=false, tailclip=false]
  n3:right:c -> n4:parent:c [headclip=false, tailclip=false]
  n4 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}4|{{<left>|<right>}}"];
  ln1l4 [style=filled, fillcolor=grey, label="NIL"];
  n4:left:c -> ln1l4:n [headclip=false, tailclip=false]
  rn1l4 [style=filled, fillcolor=grey, label="NIL"];
  n4:right:c -> rn1l4:n [headclip=false, tailclip=false]
  n5 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}5|{{<left>|<right>}}"];
  n5:left:c -> n3:parent:c [headclip=false, tailclip=false]
  n5:right:c -> n7:parent:c [headclip=false, tailclip=false]
  n6 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}6|{{<left>|<right>}}"];
  ln1l6 [style=filled, fillcolor=grey, label="NIL"];
  n6:left:c -> ln1l6:n [headclip=false, tailclip=false]
```

```

rnil6 [style=filled, fillcolor=grey, label="NIL"];
n6:right:c -> rnil6:n [headclip=false, tailclip=false]
n7 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}7|{{<left>}}|"];
n7:left:c -> n6:parent:c [headclip=false, tailclip=false]
n7:right:c -> n8:parent:c [headclip=false, tailclip=false]
n8 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}8|{{<left>}}|"];
lnil8 [style=filled, fillcolor=grey, label="NIL"];
n8:left:c -> lnil8:n [headclip=false, tailclip=false]
rnil8 [style=filled, fillcolor=grey, label="NIL"];
n8:right:c -> rnil8:n [headclip=false, tailclip=false]
n9 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}9|{{<left>}}|"];
n9:left:c -> n5:parent:c [headclip=false, tailclip=false]
n9:right:c -> n15:parent:c [headclip=false, tailclip=false]
n10 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}10|{{<left>}}|"];
lnil10 [style=filled, fillcolor=grey, label="NIL"];
n10:left:c -> lnil10:n [headclip=false, tailclip=false]
rnil10 [style=filled, fillcolor=grey, label="NIL"];
n10:right:c -> rnil10:n [headclip=false, tailclip=false]
n11 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}11|{{<left>}}|"];
lnil11 [style=filled, fillcolor=grey, label="NIL"];
n11:left:c -> lnil11:n [headclip=false, tailclip=false]
rnil11 [style=filled, fillcolor=grey, label="NIL"];
n11:right:c -> rnil11:n [headclip=false, tailclip=false]
n12 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}12|{{<left>}}|"];
n12:left:c -> n10:parent:n [headclip=false, tailclip=false]
n12:right:c -> n13:parent:n [headclip=false, tailclip=false]
n13 [style=filled, fillcolor=red, fontcolor = white, label="{{<parent>}}13|{{<left>}}|"];
n13:left:c -> n11:parent:c [headclip=false, tailclip=false]
n13:right:c -> n14:parent:c [headclip=false, tailclip=false]
n14 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}14|{{<left>}}|"];
lnil14 [style=filled, fillcolor=grey, label="NIL"];
n14:left:c -> lnil14:n [headclip=false, tailclip=false]
rnil14 [style=filled, fillcolor=grey, label="NIL"];
n14:right:c -> rnil14:n [headclip=false, tailclip=false]
n15 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}15|{{<left>}}|"];
n15:left:c -> n12:parent:n [headclip=false, tailclip=false]
n15:right:c -> n19:parent:n [headclip=false, tailclip=false]
n16 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}16|{{<left>}}|"];
lnil16 [style=filled, fillcolor=grey, label="NIL"];
n16:left:c -> lnil16:n [headclip=false, tailclip=false]
rnil16 [style=filled, fillcolor=grey, label="NIL"];
n16:right:c -> rnil16:n [headclip=false, tailclip=false]
n17 [style=filled, fillcolor=red, fontcolor = white, label="{{<parent>}}17|{{<left>}}|"];
n17:left:c -> n16:parent:c [headclip=false, tailclip=false]
n17:right:c -> n18:parent:c [headclip=false, tailclip=false]
n18 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}18|{{<left>}}|"];
lnil18 [style=filled, fillcolor=grey, label="NIL"];
n18:left:c -> lnil18:n [headclip=false, tailclip=false]
rnil18 [style=filled, fillcolor=grey, label="NIL"];
n18:right:c -> rnil18:n [headclip=false, tailclip=false]
n19 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}19|{{<left>}}|"];
n19:left:c -> n17:parent:c [headclip=false, tailclip=false]
n19:right:c -> n20:parent:c [headclip=false, tailclip=false]

```

```
n20 [style=filled, fillcolor=black, fontcolor = white, label="{{<parent>}}|20|{<left  
lnil20 [style=filled, fillcolor=grey, label="NIL"];  
n20:left:c -> lnil20:n [headclip=false, tailclip=false]  
rnil20 [style=filled, fillcolor=grey, label="NIL"];  
n20:right:c -> rnil20:n [headclip=false, tailclip=false]  
}
```