

# Arbres rouge–noir (chap. 13)

CPES2: Algorithmique et Applications

Pierre Senellart

3 décembre 2020

- Rappel sur les arbres binaires de recherche équilibrés
- Arbres rouge–noir. Propriétés des arbres rouge–noir:
  - Si un nœud est rouge, ses enfants sont noirs.
  - Même nombre de nœuds noirs sur tout chemin d’un nœud  $x$  de l’arbre aux descendants (y compris  $x$  lui-même) ayant un ou zéro enfant.
- Exemple d’arbre (voir exemple 13.4)
- Borne:  $n \geq 2^{hn(x)}$  et  $hn(x) \geq \frac{h-1}{2}$ , par récurrence sur la hauteur
- Rotation gauche, rotation droite
- Insertion dans un arbre rouge–noir:
  - Insertion dans l’arbre à l’endroit naturel  $z$ , colorié en rouge
  - Correction de la violation rouge–noir entre  $z$  et son parent:
    1. Si  $z$  est enfant de la racine, on colorie la racine en noir.
    2. On traite du cas où le parent de  $z$  est un enfant gauche, l’autre cas est symétrique.
    3. Si l’oncle de  $z$  est rouge, on le colorie ainsi que le parent de  $z$  en noir, et le grand-parent de  $z$  en rouge. On continue avec le grand-parent.
    4. Sinon, si  $z$  est un enfant droit, on fait une rotation gauche sur le parent de  $z$  et on considère le nouvel enfant gauche de  $z$  et on passe au cas suivant.
    5.  $z$  est un enfant gauche, on colorie son parent en noir et son grand-parent en rouge, puis on fait une rotation droite sur le grand-parent. Fini.
  - Illustration : insertion de 4 dans l’arbre exemple
  - Vérification des propriétés des arbres rouge–noir.
  - Complexité
- Illustration : création d’un arbre rouge–noir en insérant 41, 38, 31, 12, 19, 8
- Suppression dans un arbre rouge–noir, en deux étapes:
  1. On supprime le nœud  $z$ :
    - Si  $z$  est une feuille, on le supprime; si  $z$  était un nœud noir, on se souvient qu’il faudra corriger les couleurs au niveau de l’emplacement où était auparavant  $z$

- Si  $z$  avait un seul enfant  $x$ , on le remplace par son enfant; si  $z$  était un nœud noir, on se souvient qu'il faudra corriger les couleurs au niveau de  $x$
  - Sinon,  $z$  a deux enfants.
    - \* On recherche le nœud  $y$  de valeur minimale dans l'arbre de l'enfant droit de  $z$ . S'il est de couleur noire, on se souvient qu'il faudra corriger les couleurs au niveau de l'emplacement actuel de  $y$ .
    - \* On remplace  $y$  par son enfant droit, s'il en a un ( $y$  ne peut pas avoir d'enfant gauche).
    - \* On remplace  $z$  par  $y$ , en conservant la couleur de  $z$ .
2. S'il faut corriger les couleurs au niveau d'un emplacement  $x$ , ça veut dire qu'il manque une couleur noire au niveau de  $x$ .
- Si  $x$  est un nœud de couleur rouge, on le transforme en un nœud noir, et on a fini.
  - Si  $x$  est la racine, on a fini.
  - Sinon, on suppose que  $x$  est un enfant gauche (le cas enfant droit est symétrique):
    - a) Si le frère  $w$  de  $x$  est rouge, on le rend noir, on rend le parent  $p$  de  $w$  et  $x$  rouge, et on fait une rotation gauche au niveau de  $p$ . On se retrouve dans le cas où  $w$  est noir, qu'on traite dans l'un des trois cas ci-dessous.
    - b) Si  $w$  est noir et n'a que des enfants noirs, on rend  $w$  rouge, ce qui absorbe la couleur noire manquante au niveau de  $x$ , et on répète la procédure au niveau du parent  $p$  de  $w$  et  $x$ , auquel il manque une couleur noire.
    - c) Si  $w$  est noir et que son enfant gauche est rouge, on rend  $w$  rouge et son enfant gauche noir, puis on fait une rotation droite en  $w$ , le nouveau frère de  $x$  est un nœud noir dont l'enfant droit est rouge, et on passe au cas suivant.
    - d) Si  $w$  est noir et que son enfant droit est rouge, on colorie  $w$  dans la couleur de son parent  $p$ , on colorie  $p$  en noir et l'enfant droit de  $w$  en noir, et on fait une rotation gauche au niveau de  $p$ . Ça règle les problèmes de couleurs, on a fini.
- Illustration: suppression de 8, 12, 19, 31, 38, 41 dans l'arbre précédemment créé
  - Conclusion sur la complexité des arbres rouge-noir
  - Arbres binaires équilibrés vs tables de hachage