

# Ensembles et tableaux associatifs

## Plan de cours

Pierre Senellart

9 novembre 2017

Prendre les présences sur Moodle. Questions préliminaires ?

## 1 Introduction

- Ensembles, tableaux associatifs
- Structures de données correspondantes en Python, C++, Java
- Implémentation simple : tableau. Complexité en temps et en mémoire

## 2 Tables de hachage (11)

- Espace de hachage de taille  $m$ , fonction de hachage
- Table de hachage ; alvéoles, chaînage des collisions
- Recherche, insertion, suppression
- Complexité dans le pire des cas, sous hypothèse d'uniformité
- Conception de fonctions de hachage :
  - Par modulo (attention aux motifs)
  - Hachage universel (le nombre de fonctions t.q.  $h(k) = h(l)$  pour  $k, l$  fixés distincts est  $\leq \frac{|H|}{m}$  ; complexité
  - Choix d'un ensemble de fonction universel :  $h_{a,b} = ((ak + b) \bmod p) \bmod m$  pour  $p > m$  nombre premier,  $a \in \mathbb{F}_p \neq 0, b \in \mathbb{F}_p$
- Adressage ouvert :
  - sondage linéaire  $h(k, i) = (h'(k) + i) \bmod m$  ; inconvénients
  - double hachage  $h(k, i) = (h_1(k) + i \times h_2(k)) \bmod m$  avec  $(h_2(k), m)$  premiers entre eux ; nombre de séquences de hachage distinctes
- Tables de hachage dynamiques :
  - Doublement de la taille
  - Hachage linéaire, seulement évoqué

## 3 Arbres binaires rouge–noir (13)

- 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 nœud noirs sur tout chemin d'un nœud de l'arbre aux descendants ayant un ou zéro fils.

- Exemple 13.4 (sans le nœud  $z$ )
- Borne :  $n \geq 2^{hn(x)}$  et  $hn(x) \geq \frac{h-1}{2}$
- 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 père :
    - Si  $z$  est fils de la racine, on colorie la racine en noir
    - On traite du cas où le père de  $z$  est un fils gauche, l'autre cas est symétrique
    - Si l'oncle de  $z$  est rouge, on le colorie ainsi que le père de  $z$  en noir, et le grand-père de  $z$  en rouge. On continue avec le grand-père.
    - Sinon, si  $z$  est un fils droit, on fait une rotation gauche et on considère le nouveau fils gauche de  $z$  et on passe au cas suivant.
    - $z$  est un fils gauche, on colore son père en noir et son grand-père en rouge, puis on fait une rotation droite. Fini.
  - Complexité
- Un mot sur la suppression, pas de détails
- Arbres binaires équilibrés vs tables de hachage