

Module EA4 – Éléments d'Algorithmique II

Outils pour l'analyse des algorithmes

Dominique Poulalhon
`dominique.poulalhon@irif.fr`

Université Paris Diderot
L2 Informatique & Math-Info
Année universitaire 2019-2020

LE HACHAGE

IV. Qu'est-ce qu'une bonne fonction de hachage ?

QU'EST-CE-QU'UNE BONNE FONCTION DE HACHAGE ?

Si vous êtes adeptes de Lego, vous avez sûrement remarqué que les sachets ne ressemblent pas à ceux-là :



mais plutôt à ça :



Pourquoi ? C'est assez contre-intuitif peut-être, mais essayez donc de chercher une pièce orange 2x1 dans chaque paquet par exemple... Trouver rapidement le bon sachet n'est pas la seule chose importante, il faut aussi pouvoir le fouiller vite, et pour ça, il vaut mieux que les pièces « hachées » dans le même sachet soient le plus dissemblables possible...

QU'EST-CE-QU'UNE BONNE FONCTION DE HACHAGE ?

pour une bonne efficacité (en temps), il faut :

- trouver rapidement la bonne boîte ;
- fouiller rapidement la boîte ;

et bien sûr, il faut éviter le gâchis en espace.

la fonction de hachage doit donc

- être facile à calculer ;
- idéalement, être sans collision – c'est impossible, mais à défaut, les éléments concernés doivent être facilement discernables ;
- remplir la table *uniformément* : éviter d'avoir de grandes zones vides et de grandes zones pleines ;
- pour cela, il faut disperser les données similaires.

QU'EST-CE-QU'UNE BONNE FONCTION DE HACHAGE ?

pour répondre, il faut savoir *à quoi ressemblent les données*

Exemple

toutes les chaînes de moins de 25 caractères ne peuvent pas être des entrées d'un dictionnaire (un vrai, genre le Larousse).

Exemple

un compilateur maintient une *table des symboles* référençant les identificateurs du programme en cours de compilation
or les programmeurs ont tendance à utiliser des identificateurs qui se ressemblent :
tmp, tmp1, tmp2...

Par exemple, la fonction *h* que j'ai utilisée pour les chevaliers de la table ronde est très mauvaise : (dans une langue donnée), certaines lettres ont beaucoup plus de chance d'être l'initiale d'un nom que d'autres, et l'hypothèse de hachage uniforme simple n'est donc pas satisfaite.

en général, les données ne sont pas réparties uniformément dans l'univers des données possibles.

CONSTRUCTION DE FONCTIONS DE HACHAGE

une fonction de hachage (primaire) doit

- être facile à calculer
- remplir la table *uniformément*, donc *dispenser les données similaires*

CONSTRUCTION DE FONCTIONS DE HACHAGE

une fonction de hachage (primaire) doit

- être facile à calculer
- remplir la table *uniformément*, donc *dispenser les données similaires*

deux étapes

- transformer toute donnée en valeur numérique (entière) : cette étape est spécifique aux données considérées
- hacher les nombres : c'est cette étape qui va assurer la dispersion

CONSTRUCTION DE FONCTIONS DE HACHAGE

une fonction de hachage (primaire) doit

- être facile à calculer
- remplir la table *uniformément*, donc *dispenser les données similaires*

deux étapes

- **transformer toute donnée en valeur numérique (entière)** : cette étape est spécifique aux données considérées
- hacher les nombres : c'est cette étape qui va assurer la dispersion

pour du texte par exemple, le plus simple : remplacer chaque caractère par son code ASCII, et considérer le texte $t_0 \dots t_\ell$ comme l'entier

$$h(t_0 t_1 \dots t_\ell) = t_0 b^\ell + t_1 b^{\ell-1} + \dots + t_{\ell-1} b + t_\ell$$

(en Java : $b = 31$)

CONSTRUCTION DE FONCTIONS DE HACHAGE

deux étapes

- transformer toute donnée en valeur numérique (entière) : cette étape est spécifique aux données considérées
- **hacher les nombres** : c'est cette étape qui va assurer la dispersion

méthode *par division*

$$h(x) = x \bmod m$$

CONSTRUCTION DE FONCTIONS DE HACHAGE

deux étapes

- transformer toute donnée en valeur numérique (entière) : cette étape est spécifique aux données considérées
- **hacher les nombres** : c'est cette étape qui va assurer la dispersion

méthode *par division*

$$h(x) = x \bmod m$$

- incontestablement simple à calculer...
- mais pas de dispersion des données similaires

⇒ pas très adaptée comme fonction de hachage primaire
(mais tout à fait satisfaisante comme (base de) fonction de hachage secondaire)

CONSTRUCTION DE FONCTIONS DE HACHAGE

deux étapes

- transformer toute donnée en valeur numérique (entière) : cette étape est spécifique aux données considérées
- **hacher les nombres** : c'est cette étape qui va assurer la dispersion

méthode *par division*

$$h(x) = x \bmod m$$

méthode *par multiplication*

$$h(x) = \lfloor m \times \{Ax\} \rfloor \quad (\text{où } \{x\} = x - \lfloor x \rfloor)$$

- m a peu d'importance (par exemple une puissance de 2 ou un nombre premier conviennent)
- une bonne valeur (empirique) pour A est $\frac{\sqrt{5}-1}{2}$ (ou une approximation fractionnaire)

⇒ très bien comme fonction de hachage primaire