

Cours/TD d'algorithmique – tables de hachage

Thomas Chatain

pour accompagner le cours en visio du 15 octobre 2026

Tables de hachage – généralités

Une table de hachage est une structure de données qui permet de stocker un ensemble de couples (k, v) en visant une complexité très efficace pour les opérations suivantes :

- *insérer* (k, v)
- *rechercher* (k) : retourne la valeur associée à la clé k
- *supprimer* (k)

En ceci, les tables de hachage sont une implémentation de ce qu'on appelle un *dictionnaire*.

Contrairement à d'autres structures qui cherchent à maintenir l'ensemble des éléments triés selon les clés, ici on ne va pas chercher à comparer les clés entre elles. On va plutôt reprendre l'idée qui apparaissait dans le tri par comptage où on associait une case d'un tableau à chaque clé. Seulement, ici, on considère un univers de clés U potentiellement très grand ; il est donc hors de question de réserver un tableau qui contienne une case distincte pour chaque clé de U . En revanche, on s'attend à ce que très peu des clés de U soient effectivement utilisées. Chaque case de la table de hachage va donc recevoir un ensemble de clés de U , en espérant que seulement quelques unes de ces clés seront effectivement utilisées en même temps. Les couples (k, v) présents dans une case de la table sont simplement stockés dans une liste chaînée.

Ainsi, une table de hachage comporte :

- un tableau T de m cases contenant chacune une liste de couples (k, v) . Pour une fois, on numérottera ces cases de 0 à $m - 1$ (ce sera plus pratique pour l'arithmétique modulaire).
- une *fonction de hachage* $h : U \rightarrow \{0, \dots, m - 1\}$ qui attribue à chaque clé k de U , la case dans laquelle sera rangé un couple (k, v) .

Ainsi, pour rechercher la valeur associée à une clé k , il suffit de calculer $h(k)$ et de parcourir la liste de couples présents dans $T[h(k)]$. La suppression fonctionne de la même façon, et l'insertion est évidente.

Bien sûr, comme U est très grand, on aura $m \ll |U|$ et h sera très loin d'être injective. Lorsque deux couples (k, v) et (k', v') sont présents dans la même case $T[i]$, c'est que $i = h(k) = h(k')$. On parle de *collision*. L'essentiel du travail sur les tables de hachage va constituer à limiter le nombre de collisions afin que les listes de couples restent très courtes (puisque, à part le parcours des listes, tout se fait en temps constant).

Familles universelles de fonctions de hachage

Afin donc de limiter les collisions, on va faire confiance au hasard pour trouver une bonne fonction de hachage. Idéalement, on pourrait choisir une fonction de hachage au hasard parmi l'ensemble $\{0, \dots, m - 1\}^U$, mais le fait de tirer une telle fonction serait excessivement long (il faudrait tirer au hasard $h(k)$ pour chaque $k \in U$), et ensuite il faudrait stocker/représenter la fonction obtenue, ce qui n'est pas réaliste.

On va donc se restreindre à une famille H de fonctions de hachage faciles à représenter, qui vont se comporter aussi bien pour les collisions que si l'on avait tiré une fonction dans $\{0, \dots, m - 1\}^U$:

Définition 1 Une famille H de fonctions de hachage de $U \rightarrow \{0, \dots, m - 1\}$ est dite universelle si, pour toute paire $\{k, k'\}$ de clés distinctes, la probabilité que h provoque la collision $h(k) = h(k')$

quand on tire h uniformément dans H , est inférieure ou égale à $\frac{1}{m}$, qui est exactement la probabilité de collision que l'on aurait eue si l'on avait tiré h uniformément dans $\{0, \dots, m-1\}^U$.

Nous allons maintenant montrer une construction qui donne des familles universelles.

Propriété 1 Soit p un grand nombre premier. La famille

$$H \stackrel{\text{def}}{=} \{h_{a,b} \mid a \in \{1, \dots, p-1\}, b \in \{0, \dots, p-1\}\}$$

avec $h_{a,b}(k) \stackrel{\text{def}}{=} g_{a,b}(k) \bmod m$ et $g_{a,b}(k) \stackrel{\text{def}}{=} ak + b \bmod p$, est une famille universelle.

Intuitivement : la fonction $g_{a,b}$ envoie de façon pseudo-aléatoire k dans $\mathbb{Z}/p\mathbb{Z}$ (qui est un grand ensemble puisque p est grand), puis $h_{a,b}$ “replie” $\mathbb{Z}/p\mathbb{Z}$ sur $\{0, \dots, m-1\}$.

Preuve. Soient k, k' deux clés distinctes. Nous cherchons les valeurs de a et b qui mettent k et k' en collision :

$$\begin{aligned} h_{a,b}(k) = h_{a,b}(k') &\iff g_{a,b}(k) \equiv g_{a,b}(k') \pmod{m} \\ g_{a,b}(k) - g_{a,b}(k') &\equiv 0 \pmod{m} \\ g_{a,b}(k) - g_{a,b}(k') &\in \{0, m, 2m, \dots, \left\lfloor \frac{p-1}{m} \right\rfloor m\} \end{aligned}$$

Or $g_{a,b}(k) - g_{a,b}(k') = (a(k-k')) \bmod p$. On doit donc résoudre dans $\mathbb{Z}/p\mathbb{Z}$ les équations $a(k-k') = 0$, $a(k-k') = m$, $a(k-k') = 2m, \dots$. Comme $k \neq k'$ et p est grand, $k-k'$ est inversible dans $\mathbb{Z}/p\mathbb{Z}$; notons l son inverse. On a donc les solutions : $a = 0l$, $a = ml$, $a = 2ml, \dots$, $a = \left\lfloor \frac{p-1}{m} \right\rfloor ml$. Seulement $a = 0$ n'est pas utilisée dans notre famille H ; il reste donc $\left\lfloor \frac{p-1}{m} \right\rfloor$ valeurs possibles pour a . On remarque que b a disparu de l'équation, ce qui signifie que pour chaque a toutes les valeurs de b conviennent. On a donc $\left\lfloor \frac{p-1}{m} \right\rfloor \times p$ couples (a, b) solutions de notre équation, soit autant de fonctions $h_{a,b}$ qui mettent k et k' en collision. Or le nombre total de fonctions de hachage dans notre famille H est $|H| = (p-1) \times p$, donc $\left\lfloor \frac{p-1}{m} \right\rfloor \times p \leq \frac{|H|}{m}$. La probabilité de collision est donc inférieure à $\frac{1}{m}$.

Double hachage

On cherche un moyen de stocker un ensemble de n couples (k, val) dans une structure composée d'une table de hachage primaire T (avec fonction de hachage h) qui pointe vers n tables secondaires $T_0 \dots T_{n-1}$ (on note h_i la fonction de hachage de T_i) de telle sorte qu'un couple (k, val) est stocké dans la table secondaire T_i avec $i = h(k)$, et son indice j dans cette table secondaire est calculé par la fonction de hachage h_i de la table T_i par $j = h_i(k)$. Ainsi, le couple (k, val) est stocké dans $T_i[h_i(k)]$ avec $i = h(k)$, ce qui permet un accès en temps constant.

La question est, pour un ensemble fixé (statique) de n couples (k, val) à stocker, de trouver des fonctions de hachage h et $h_0 \dots, h_{n-1}$ qui évitent toute collision dans les tables secondaires.

Pour cela, on opère en deux étapes : on va d'abord chercher une fonction de hachage h qui limite le nombre de collisions, c'est-à-dire le nombre de paires $\{k, k'\}$, $k \neq k'$, telles que $h(k) = h(k')$.

On remarque que, la table primaire ayant exactement n cases, il serait en principe possible de n'obtenir aucune collision, mais la probabilité de tirer une telle fonction h est extrêmement faible. On va donc se contenter d'une fonction h qui limite le nombre de collisions, et utiliser les petites tables secondaires pour stocker les couples.

On rappelle qu'une famille H de fonctions de hachage pour une table de taille m est *universelle* si, quand on tire aléatoirement (avec probabilité uniforme) une fonction h dans H , la probabilité que h provoque la collision $h(k) = h(k')$, est inférieure ou égale à $\frac{1}{m}$.

Question 1

Montrer que, si on tire une fonction de hachage h uniformément dans une famille universelle pour une table de hachage de taille m destinée à stocker n couples (k, val) , alors l'espérance du nombre total de collisions obtenues est inférieure ou égale à $\frac{n(n-1)}{2m}$.

Question 2

Afin d'éviter toute collision dans les tables secondaires T_i , on leur alloue des plages m_i suffisamment grandes : on choisit $m_i = n_i^2$ où n_i est le nombre de couples à stocker dans la table T_i .

Montrer que, si on tire une fonction de hachage h_i uniformément dans une famille universelle H_i pour une table de hachage de taille $m_i = n_i^2$, alors l'espérance du nombre total de collisions obtenues est inférieure ou égale à $\frac{1}{2}$.

Question 3

En déduire qu'au moins la moitié des fonctions h_i de H_i ne provoquent aucune collision parmi les n_i couples à stocker dans la table T_i , et que le nombre moyen de tirages aléatoires uniformes à effectuer pour trouver une telle fonction h_i est constant.

Question 4

Établir une relation entre la somme des tailles m_i de toutes les tables secondaires et le nombre de collisions provoquées par la fonction de hachage primaire h appliquée sur les n couples à stocker.

Question 5

Montrer que, si l'on tire aléatoirement une fonction de hachage primaire h , puis que l'on construit des tables secondaires sans collisions comme indiqué ci-dessus, l'espérance de l'espace total utilisé par les tables est $O(n)$.

Note sur l'implémentation

En pratique, la table primaire devra contenir des pointeurs vers les tables secondaires T_i , mais aussi les tailles m_i associées et les paramètres a_i, b_i qui définissent les h_i par

$$h_i(k) = ((a_i k + b) \mod p) \mod m_i$$

avec un entier premier $p \gg n$.