TD 7 : Tables de hachage

17 Octobre 2024

Page web du cours : `https://people.irisa.fr/David.Baelde/algo/index.html`
Coordonnées des chargés de TD : `malo.revel@ens-rennes.fr` `isseinie.calviac@irisa.fr`
Les TDs ont lieu les jeudis de 9h45 à 11h15, se fier à ADE pour la salle.

1 Utilisation d'une table de hachage

Question 1 – Soit une table de hachage T par chaînage à M alvéoles, dont la fonction de hachage h est uniforme sur les entrées considérées. On ajoute n valeurs à T en les tirant aléatoirement et équiprobablement dans $[\![0, N-1]\!]$ pour un certain $N > 0$.

— Quelle est la complexité temporelle au pire cas du test d'appartenance d'une clef k à T ?
— Quelle est la complexité moyenne du test d'appartenance d'une clef k à T ?
— Quelle est la complexité moyenne de l'ajout d'une clef k à T ?

— $O(n)$ (dans le pire cas tous les éléments sont dans la même alvéole)
— Pour toute alvéole j, d'après le cours l'espérance du nombre d'éléments dans l'alvéole j est inférieure à $\frac{n}{M}$. Donc $O(\frac{n}{M})$.
— $O(1)$ (c'est un ajout dans une liste chaînée)

Question 2 – On considère le problème suivant :

Entrée : une liste R de couples d'entiers représentant une relation binaire.

Sortie : un booléen indiquant si R est une relation symétrique.

On suppose que l'on dispose d'une fonction de hachage $h : \mathbb{N}^2 \to [\![0, M-1]\!]$ uniforme sur les entrées du problème. Donner un algorithme qui résoud le problème avec une complexité temporelle moyenne $O(|R|^2/M)$.

Fonction $\mathrm{SYM}(R)$
  T ← Table de hachage de taille M, avec comme fonction de hachage h
  pour tout élément (a, b) dans R **faire**
    Ajouter (a, b) dans T

pour tout élément (a, b) dans R **faire**
  si (b, a) n'est pas présent dans T **alors**
    return false
return true

Coût de la première boucle : en notant α_j (c'est une variable aléatoire) le nombre de clefs dans l'alvéole j à la fin de l'exécution de la première boucle, on peut majorer le coût de la première boucle par $O(\mathbb{E}(\sum_{(a,b)\in R} \alpha_{h(a,b)})) = O(\sum_{(a,b)\in R} \mathbb{E}(\alpha_{h(a,b)}))$ (car on ne fait que rajouter des éléments).

Or (propriété du cours) comme h est uniforme, pour tout j, $\mathbb{E}(\alpha_j) \leq \frac{|R|}{M}$. Donc le coût de la première boucle (et de la deuxième, et de tout l'algorithme) est $O(\sum_{(a,b)\in R} \frac{|R|}{M}) = O(\frac{|R|^2}{M})$.

2 Hachage universel

Question 3 – Montrer que la famille de fonctions de hachage constantes suivante (vue en cours) est uniforme, mais pas universelle :

Soit M un nombre d'alvéoles.

1. On tire un indice $j \in [\![0, M-1]\!]$ uniformément.
2. On prend $h_j : \begin{array}{ccc} \mathbb{K} & \to & [\![0, M-1]\!] \\ x & \mapsto & j \end{array}$ comme fonction de hachage

Uniforme : La fonction de hachage h est bien uniforme car $\mathbb{P}(h(K) = j) = \mathbb{P}(h = h_j) = \frac{1}{M}$.
Pas universelle : $\mathbb{P}(h(k) = h(l)) = 1$

Question 4 – Montrer que la famille de fonctions de hachage suivante est universelle, mais pas uniforme :

Soit p un nombre premier.

1. on choisit uniformément $a = (a_1, \ldots, a_r) \in \{0, \ldots, p-1\}^r$;
2. on prend $h_a : \begin{array}{ccc} \mathbb{K} & \to & \mathbb{Z}/_{p\mathbb{Z}} \\ x & \mapsto & \sum_{i=1}^{r} a_i x_i \mod p \end{array}$ comme fonction de hachage.

On considère ici que $M = p$.
Indice : comme p est premier, $\mathbb{Z}/_{p\mathbb{Z}}$ est un corps.

Universelle :
Soit $x \neq y$ deux clefs différentes. Montrons que $\mathbb{P}(h_a(x) = h_a(y)) \leq \frac{1}{p}$. Comme $x \neq y$, il existe un indice $j \in \{1, \ldots, r\}$ tel que $x_j \neq y_j$.

$$h_a(x) = h_a(y) \text{ ssi } \sum_{i=1}^{r} a_i x_i = \sum_{i=1}^{r} a_i y_i \mod p$$
$$\text{ssi } a_j \underbrace{(y_j - x_j)}_{z} = \underbrace{\sum_{i \neq j} a_i (x_i - y_i)}_{m} \mod p$$

On a $z \neq 0$. L'équation $a_j z = m \mod p$ admet une seule solution a_j dans $\{0, \ldots, p-1\}$. C'est parce que $\mathbb{Z}/_{p\mathbb{Z}}$ est un corps, la solution s'écrit $a_j = mz^{-1}$.
Donc $\mathbb{P}(h_a(x) = h_a(y)) = \frac{1}{p}$.

Pas uniforme : pour $k = 0$, on a $h(k) = 0$ avec probabilité 1 et 0 pour les autres indices.

3 Recherche de motif avec hachage

Soit Σ un alphabet fini. On supposera que $\Sigma = \{0, \ldots, |\Sigma| - 1\}$. Étant donné un texte t de longueur $|t| = n > 0$ et un motif m de longueur $|m| = k > 0$, on cherche à déterminer si m est un sous-mot de t, c'est-à-dire qu'on cherche $i \in [\![0, n-k]\!]$ tel que $t[i..i+k-1] = m$, où $t[d..f]$ est le sous-mot de t de l'indice d (inclus) à l'indice f (inclus).

Question 5 – Proposer un algorithme naïf résolvant ce problème. Donner et justifier sa complexité en temps dans le pire cas.

Fonction NAIF(t de longueur n, m de longueur k)
  pour $i \leftarrow 0$ à $n-k$ **faire**
    $c \leftarrow$ true
    pour $j \leftarrow 0$ à $k-1$ **faire**
      si $t[i+j] \neq m[j]$ **alors**
        $c \leftarrow$ false
        break
    si c **alors return** true
  return false

Complexité : $O((n-k).k)$, atteinte pour $t = a^n$ et $m = a^{k-1}b$ avec $\Sigma = \{a, b\}$ (ou $t = a^{n-1}b$ si on veut une instance positive).

On considère maintenant $h : \Sigma^* \to [\![0, p-1]\!]$ avec $p > 0$, une fonction de hachage telle que :

$$\forall l \in \mathbb{N}, \forall i \in [\![0, n-l-1]\!],\ h(t[i+1..i+l]) = h(t[i..i+l-1]) - h(t[i]) + h(t[i+l]) \mod p$$

Question 6 – Donner un exemple d'une telle fonction de hachage.

$$h(t[i..i+l-1]) = \sum_{j=0}^{l-1} t[i+j] \mod p$$

QUESTION 7 – Proposer un algorithme de recherche de motif qui utilise cette propriété en hachant des sous-mots de t de taille k. Donner et justifier sa complexité en temps dans le pire cas, en supposant que les opérations sur les hachés se font en temps $O(1)$, et que le hachage d'un sous-mot de taille l se fait en temps $O(l)$.

Fonction RABIN-KARP(t de longueur n, m de longueur k)
  $hm \leftarrow h(m[0..k-1])$
  $ht \leftarrow h(t[0..k-1])$
  pour $i \leftarrow 0$ **à** $n-k+1$ **faire**
    si $ht = hm$ **alors**
      si $t[i..i+k-1] = m$ **alors**
        return true
    $ht \leftarrow ht + h(t[i+k]) - h[i]$
  return false

Le test $ht = hm$ et la mise à jour de ht se font en temps $O(1)$.

Dans le pire cas, le test $ht = hm$ est toujours vérifié, et la comparaison $t[i..i+k-1] = m$, en temps $O(k)$, est réalisée à chaque itération. Au pire cas la complexité de cet algorithme est donc $O(k + k + (n-k).k) = O((n-k).k)$, qui n'est pas meilleure que celle de l'algorithme naïf.

QUESTION 8 – On propose maintenant la fonction de hachage suivante :

$$h(t[i..i+l-1]) = \sum_{j=0}^{l-1} B^{l-1-j}.t[i+j] \mod p$$

avec $B \in \mathbb{N} \setminus \{0\}$.

Comment calculer efficacement $h(t[i+1..i+k])$ à partir de $h(t[i..i+k-1])$? Adapter l'algorithme précédent avec cette nouvelle fonction de hachage. On suppose que chaque multiplication se fait en temps $O(1)$.

$$\begin{aligned} h(t[i+1..i+k]) &= \sum_{j=0}^{k-1} B^{k-1-j}.t[i+1+j] \\ &= \sum_{j=1}^{k} B^{k-j}.t[i+j] \\ &= \sum_{j=0}^{k-1} B^{k-j}.t[i+j] - B^k.t[i] + B^0.t[i+k] \\ &= B.(h(t[i..i+k-1]) - B^{k-1}.t[i]) + t[i+k] \end{aligned}$$

En calculant au préalable $B^{k-1} \mod p$ (coût $O(k)$), cette opération de shift se fait en $O(1)$ si on connaît $t[i..i+k-1]$.

Pour adapter l'algorithme on ajoute au début le calcul de B^{k-1}, et on remplace la ligne de mise à jour de ht par

$$ht \leftarrow B.(ht - B^{k-1}.t[i]) + t[i+k]$$

La complexité ne change pas (le calcul de B^{k-1} et le hachage de $t[0..k-1]$ et m sont négligeables).

QUESTION 9 – (Bonus) Évaluer empiriquement cette deuxième version de l'algorithme (en comptant le nombre de collisions) en prenant pour p un entier premier le plus grand possible et $B = 256$, pour t le roman de votre choix et pour m le mot de votre choix.

Remarque : bien que la complexité temporelle pire cas de cet algorithme ne soit pas meilleure que celle de l'algorithme naïf, en pratique sur des exemples réalistes le nombre de collisions est faible et on se rapproche d'une complexité linéaire en $n + k$.

4 Hachage parfait

Nous avons une grande liste de n mots fixés et nous souhaitons une fonction de hachage sans collision sur ces mots, afin de les utiliser comme clés dans un dictionnaire efficace.

4.1 Étape 1 : table sans collision, en taille $O(n^2)$

Soit $M = n^2$. Prenons une fonction de hachage $H \to \{0, ..., M-1\}$ choisie aléatoirement selon l'hypothèse d'universalité. Commençons par estimer le nombre de collisions après l'insertion de n clés distinctes.

QUESTION 10 – Soit X le nombre de collisions obtenues avec H. Montrer que $\mathbb{E}(X) = \frac{n(n-1)}{2M}$.

Notons X le nombre de collisions obtenues avec H. On a :

$$X = \sum_{\{k,\ell\} \text{ où } k,\ell \text{ dans la table } |k\neq\ell} 1_{H(k)=H(\ell)}.$$

En effet, on compte un pour chaque ensemble $\{,\ell\}$ de clefs distinctes quand elle forme une collision, i.e. quand $H(k) = H(\ell)$. Par linéarité de l'espérance on a :

$$\mathbb{E}(X) = \sum_{\{k,\ell\} \text{ où } k,\ell \text{ dans la table } |k\neq\ell} \mathbb{E}(1_{H(k)=H(\ell)})$$

Mais

$$\begin{aligned} \mathbb{E}(1_{H(k)=H(\ell)}) &= \mathbb{P}(H(k) = H(\ell)) \\ &\leq \tfrac{1}{M} \qquad \text{car } H \text{ est universelle.} \end{aligned}$$

On a :

$$\mathbb{E}(X) = \sum_{\{k,\ell\} \text{ où } k,\ell \text{ dans la table } |k\neq\ell} \mathbb{E}(1_{H(k)=H(\ell)}) \leq \sum_{\{k,\ell\} \text{ où } k,\ell \text{ dans la table } |k\neq\ell} \frac{1}{M} \leq \binom{n}{2}\frac{1}{M}.$$

Question 11 – Montrer que $\mathbb{P}(X = 0) > \frac{1}{2}$

Indice : l'inégalité de Markov nous dit que $\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}(X)}{t}$.

$$\mathbb{E}(X) \leq \frac{n(n-1)}{2}\frac{1}{n^2} < \frac{1}{2}.$$

Ici, pour $t = 1$ l'inégalité de Markov donne $\mathbb{P}(X \geq 1) \leq \frac{1}{2}$. Donc $\mathbb{P}(\text{non } X \geq 1) = \mathbb{P}(X = 0) > \frac{1}{2}$. Autrement dit, $\mathbb{P}(\text{pas de collision avec } H) > \frac{1}{2}$.

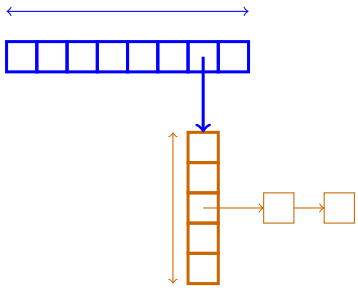
Question 12 – Supposons que nous tirons une nouvelle fonction de hachage, tant que nous obtenons une collision sur notre ensemble de clés. Quelle est l'espérance du nombre de tirages avant de trouver une fonction sans collisions ? Conclure.

Indice : $\sum_{k=0}^{\infty} k(1-p)^{k-1} = 1/p^2$

4.2 Étape 2 : . . . puis en taille $O(n)$

Construisons une structure à deux niveaux :

1. Premier niveau avec $M = n$ alvéoles, avec une fonction de hachage h. Il peut y avoir des collisions.
2. Dans l'alvéole numéro j du premier niveau, on met une table de hachage sans collision T_j. On utilise l'étape 1 : on met ${a_j}^2$ sous-alvéoles dans T_j où a_j = le nombre d'éléments dans l'alvéole numéro j.



On va montrer que l'on peut choisir h du premier niveau tel que l'espace du deuxième niveau $\sum_{j=0}^{M-1} {a_j}^2 = O(n)$.

QUESTION 13 – Posons $M = n$. Soit H une fonction de hachage aléatoire universelle.

Soit ${a_j}^H$ la variable aléatoire qui représente le nombre d'éléments dans l'alvéole numéro j au premier niveau avec H. Montrer que :

$$\mathbb{E}(\sum_{j=0}^{M-1} \left({a_j}^H\right)^2) < 2n.$$

Indice : pour tout a entier, $a^2 = a + 2(a(a-1)/2)$.

QUESTION 14 – En déduire que $\mathbb{P}(\sum_{j=0}^{M-1} \left({a_j}^H\right)^2 < 4n) > \frac{1}{2}$.

QUESTION 15 – Comment construire une table de hachage avec un accès en temps $O(1)$ au pire cas et un coût en espace mémoire en $O(n)$?