

HACHAGE

Hachage

TAD :

opérations $\text{insert}(k)$, $\text{delete}(k)$, $\text{find}(k)$ où $k \in U$ est une **clé**

(U : univers de clés)

Tableau de hachage : taille M

on veut y placer N éléments

facteur de remplissage (*load factor*) $\alpha = N/M$

Fonction de hachage

Fonction de hachage : $h: U \mapsto \{0, 1, \dots, M - 1\}$

collision : $h(k) = h(k')$ mais $k \neq k'$

Birthday paradox (clés aléatoires)

Probabilité d'aucune collision

$$p = 1 \left(1 - \frac{1}{M}\right) \left(1 - \frac{2}{M}\right) \cdots \left(1 - \frac{N-1}{M}\right)$$
$$< \prod_{i=0}^{N-1} e^{-i/M} = \exp\left(-\frac{(N-1)N}{2M}\right)$$

$N > \sqrt{M2 \ln 2} \approx 1.18\sqrt{M}$, \Rightarrow au moins une collision avec probabilité $\geq 1/2$.

Fonctions de hachage

on veut éviter les collisions : disperser les éléments autant que possible

idéalement : $h(k)$ a la distribution uniforme sur $\{0, \dots, M - 1\}$.

en pratique : on ne sait pas la distribution de k

Fonctions de hachage (cont)

Méthode de la division : $h(k) = k \bmod M$

Choix de M : on veut éviter la réduction de l'espace de valeurs de hachage à cause des données non-aléatoires

- ne devrait pas être $M = 2^k$ (les derniers k bits de la clé déterminent la valeur de hachage)
- puissances de 10 à éviter (nombres décimaux)
- ne devrait pas être un multiple de 3 (permutations d'une clé binaire donnent des valeurs de hachage qui diffèrent par multiples de 3)

Règle : choisir un prime qui est loin de 2^k et 10^m .

Fonctions de hachage (cont)

Méthode de la multiplication : $h(k) = \left(\lfloor M\{\gamma k\} \rfloor \right)$

où w est la longueur d'un mot-machine et $\{x\} = x - \lfloor x \rfloor$

Choix de M : $M = 2^p$ (p bits de l'ordre supérieur de la moitié inférieure de Ak)

Poser $\gamma = \frac{A}{2^w}$

Choix de A : prime relatif à 2^w

- Ak devrait être distribué «uniformement»

$A \approx \frac{\sqrt{5}-1}{2} 2^w$ est un bon choix (le prime relatif le plus proche)

Hachage universel

Choisir une fonction de hachage **aléatoire** :

écrire la clé de $(r + 1)$ «caractères» comme $x = \langle x_0, x_1, \dots, x_r \rangle$ et calculer

$$h(x) = \left(\sum_{i=0}^r h_i(x_i) \right) \bmod M$$

où h_i sont aléatoires

et M est un prime

P.e., $h_i(x) = a_i x \bmod M$, $a_i \in \{0, \dots, M - 1\}$ aléatoire uniforme

Collision entre $x = \langle x_0, \dots, x_r \rangle$ et $y = \langle y_0, \dots, y_r \rangle$:

$$\sum_{i=0}^r a_i(x_i - y_i) \equiv 0 \pmod{M}$$

Il existe M^r solutions pour $\langle a_0, \dots, a_r \rangle$

\Rightarrow probabilité de collision $\frac{M^r}{M^{r+1}} = \frac{1}{M}$. (indépendamment de x, y)

Résolution de collisions

1. chaînage : stocker une liste pour chaque valeur de hachage
2. adressage ouvert : séquence de sondage

Chaînage

Liste chaînée

Performance : longueur d'une liste en moyenne est α

Nombre de comparaisons pour recherche avec succès $1 + \alpha/2$

Permet $\alpha > 1$

Adressage ouvert

Sondage d'une séquence de positions : dépend de la clé à insérer

On essaie les cases $h_0(k), h_1(k), \dots$

Avec une fonction $f : h_i(k) = h(k) + f(i, k) \bmod M$

f : stratégie de résolution de collision

Méthodes :

- sondage linéaire $f(i, k) = i$
- sondage quadratique $f(i, k) = i^2$
- double hachage $f(i, k) = ih'(k)$ avec fonction de hachage auxiliaire h'

Ne permet pas $\alpha > 1$

Sondage linéaire

(*Linear probing*) : $h(k), h(k) + 1, h(k) + 2, \dots$

Problème : grappe forte (*primary clustering*) — blocs de cases occupées

Nombre moyen de sondages lors de recherche fructueuse : $\approx \frac{1}{2} \left(1 + \frac{1}{1-\alpha} \right)$

sondages lors de recherche infructueuse ou insertion : $\approx \frac{1}{2} \left(1 + \frac{1}{(1-\alpha)^2} \right)$

\Rightarrow utiliser jusqu'à $\alpha < 0.75$

Sondage quadratique

(*Quadratic probing*) : $h(k), h(k) + 1, h(k) + 4, h(k) + 9, \dots$

Problème : grappe faible (*secondary clustering*) — suite de cases occupées par clés avec la même valeur de hachage

N'est pas certain que l'on trouve une case vide lors du sondage que si $\alpha < 0.5$

Double hachage

Généralisation de sondage linéaire : $h(k), h(k) + c, h(k) + 2c, h(k) + 3c, \dots$
 c dépend de la clé k : $c = h'(k)$

Exemples (éviter $c = 0$!) : $h'(x) = 1 + x \bmod M'$ avec $M' < M$
 $h'(x) = M' - (x \bmod M')$

Très proche d'une résolution idéale

Nombre moyen de sondages lors d'insertion ou recherche infructueuse :
 $\approx 1 + \alpha + \alpha^2 + \dots \approx \frac{1}{1-\alpha}$

sondages lors de recherche fructueuse :
 $\approx \frac{1}{N} \sum_{i=0}^{N-1} \frac{1}{1-i/M} \approx \frac{1}{\alpha} \ln \frac{1}{1-\alpha}$