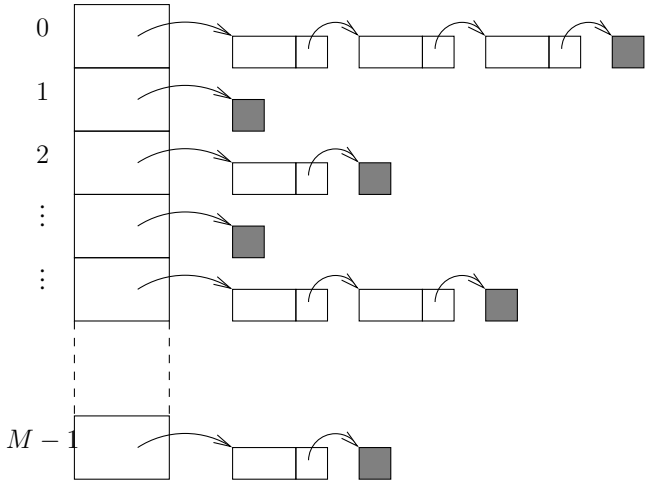
Table de hachage

Concepts : Recherche d'éléments, test d'égalité
Méthodes : Adressage direct, Fonction de hachage

Adressage fermé

On considère le problème de hachage fermé dans une table de taille M. La fonction de hachage est notée h et le nombre de clés à "hacher" est N. Dans ce TD on considère un hachage fermé. c'est à dire que la résolution de collision se fait par chaînage externe à la table.



On suppose que la fonction de hachage h est uniforme. Ceci se modélise en associant à chaque clé x de l'espace des clés une variable aléatoire U_x de loi uniforme à valeurs dans $\{0, \cdots, M-1\}$. On suppose que ces variables aléatoires sont indépendantes entre elles. Pour notre problème, on notera les N variables aléatoires $\{U_1, U_2, \cdots, U_N\}$.

L'état de remplissage de la table est modélisé par un vecteur de dimension M : $X = [X_0, \cdots, X_{M-1}]$, où X_i représente le nombre de clés hachées sur la case i.

Question 1 : Calcul des X_i

Exprimer les X_i en fonction des U_i. Pour i donné calculer la loi de X_i.

Question 2 : Recherche infructueuse

Calculer le coût moyen d'une recherche d'une clé qui n'est pas dans la table. On notera $\alpha = \frac{N}{M}$ le taux de remplissage de la table.

Question 3 : Recherche d'un élément de la table

Calculer le coût moyen d'une recherche d'une clé que l'on sait dans la table.

Question 4 : Insertion et suppression d'un élément de la table

Calculer le coût moyen d'une insertion d'une nouvelle clé dans la table. Calculer le coût moyen d'une suppression d'une clé de la table.

Question 5 : Taux de cases vides

Hachage

Calculer la probabilité qu'une case soit vide, donner un équivalent en fonction de α pour N et M grands. En déduire le nombre moyen de cases vides dans la table.

Question 6 : Approximation

On pourra vérifier (à la maison) que pour n pas trop grand et M et N grands

$$\mathbb{P}(X_i = n) \simeq e^{-\alpha}\frac{\alpha^k}{k!}.$$

Approximation par une loi de Poisson.

En utilisant l'inégalité

$$e^{\alpha} \leqslant 1 + \frac{\alpha}{1!} + \frac{\alpha^2}{2!} + \cdots + \frac{\alpha^k}{k!} + \frac{\alpha^{k+1}}{(k+1)!}e^{\alpha},$$

montrer que

$$\mathbb{P}(X_i > k) \leqslant \frac{\alpha^{k+1}}{(k+1)!}$$

En déduire que

$$\mathbb{P}(X_i > b\alpha) \leqslant (\frac{e}{b})^{b\alpha}.$$

On pourra utiliser la formule de Stirling $n! \simeq \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$.

Comment interpréter cette quantité ? Pourquoi peut-on parler de bon comportement de la recherche en table ?

Calculer pour différentes valeurs de b et de α cette probabilité.

Panini

Votre petit cousin collectionne les vignettes autocollantes des footballeurs célèbres. Il achète au marchand de journaux des pochettes de 10 vignettes à un prix de 2 euros. L'album de la collection complète contient $M = 300$ places pour les vignettes autocollantes (prix d'achat de l'album 4 euros). Il souhaite savoir combien de pochettes il devra acheter pour avoir toute la collection. Saurez-vous lui répondre ?

On simplifie le modèle en supposant que l'on achète les vignettes une par une et on note T le nombre total de vignettes achetées pour avoir toute la collection. A la fin, il restera plein de doubles ! On supposera également que l'éditeur des vignettes est honnête et qu'il imprime les vignettes dans les mêmes proportions (il n'y a pas de vignettes plus rares que d'autres).

Lorsque la collection comporte déjà $i-1$ vignettes, on note Y_i le nombre de vignettes à acheter pour avoir une vignette qui n'est pas dans la collection.

Question 1 :

Donner, en la justifiant, la loi de Y_i. Les variables Y_i sont-elles indépendantes.

Question 2 :

Exprimer T en fonction des Y_i. En déduire $\mathbb{E}T$ et $VarT$. Commenter votre résultat et répondre à votre petit cousin.

Question 3 :

Quel est le rapport entre les panini et les tables de hachage ?

Hachage

Algorithme de Karp-Rabin

On veut rechercher un motif M de longueur m dans un texte T de longueur n. Les textes sont exprimés dans un alphabet de K lettres. Le texte et le motif seront donnés dans deux tableaux, ces tableaux seront indicés à partir de 1.

Question 1 : Test d'égalité de mots

Écrire un algorithme qui teste l'égalité de deux mots. Calculer le coût de votre algorithme en nombre de comparaisons de caractères.

Afin de rechercher plus efficacement un motif, on dispose d'une fonction de hachage H sur les mots. Si A est un tableau de lettres $h(A, i, j)$ représente la valeur de hachage du sous-mot du tableau A défini de l'indice i à j inclus. On suppose que la fonction h prend des valeurs entières avec $0 \leqslant h(A, i, j) \leqslant H-1$.

```
SOUS-CHAINE(T,M)
  Données : T texte dans lequel le motif M est recherché
  Résultat : Écrit la liste des indices où démarrent le motif M dans le texte T
  n = taille(T)
  m = taille(M)
  hmotif = h(M[1..m])
  hcourant = h(T[1..m])
  for i = 1 à n − m + 1
      if (hcourant=hmotif)
          // Commentaire 1 :.....................................
          if (égalité(T[i, i + m − 1], M)
              // Commentaire 2 :.....................................
              Écrire i
      hcourant = h(T[i + 1..i + m])
      // Commentaire 3 :.....................................
```

Question 2 : Commentaires

Écrire les commentaires 1, 2 et 3 et calculer le coût maximal de cet algorithme en nombre de comparaisons de caractères.

Question 3 : Propriétés de h

Quelles sont les propriétés que h doit vérifier pour que cet algorithme soit efficace en moyenne ? Calculer dans ce cas le coût moyen de cet algorithme.

On suppose que chaque caractère est codé par un entier et que la fonction h s'écrit pour un mot $A = (a_1, \cdots, a_m)$

$$h(A[1..m]) = (a_1 + a_2 + \cdots + a_m) \text{ modulo } p,$$

avec p un entier donné.

Question 4 : h additive

Expliquer le rôle de l'entier p. Comment calculer $h(T[i..i + m - 1])$ en fonction de $h(T[i - 1..i + m - 2])$, en déduire que la ligne 13 de l'algorithme ne nécessite que deux consultations du tableau T.

Hachage

On se donne l'alphabet $\{A, B, C, D\}$ avec le codage standard $\{0, 1, 2, 3\}$. Soit

$$T = ABCADBBACABAC,\ M = CAB.$$

Question 5 : Exemple

Dérouler l'algorithme pour $p = 9$ et évaluer le nombre de comparaisons de caractères effectuées. Quel inconvénient voyez-vous à cette fonction h ?

On considère maintenant une nouvelle fonction h_1 construite sur le mot A par

$$h_1(A[1..m]) = (a_1.d^{m-1} + a_2.d^{m-2} + \cdots + a_{m-1}.d^1 + a_m.d^0) \text{ modulo } p,$$

avec d un entier donné (base).

Question 6 : h polynomiale

Expliquer le rôle de l'entier d. Comment calculer $h(T[i..i+m-1])$ en fonction de $h(T[i-1..i+m-2])$, en déduire que la ligne de mise à jour de $hcourant$ de l'algorithme ne nécessite que deux consultations du tableau T.

Question 7 : Exemple

Dérouler l'algorithme sur le même exemple ci-dessus pour $p = 9$ et $d = 4$. Évaluer le nombre de comparaisons de caractères effectuées. Commenter votre résultat.