

Examen

UPMC — Master d'Informatique — STL - M1
UE Algorithmique avancée

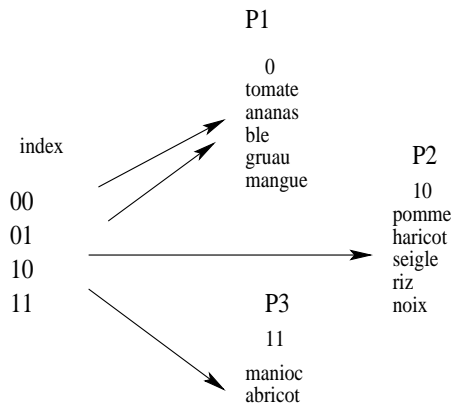
21 janvier 2005

1 Hachage extensible [10 points]

Dans la méthode de hachage extensible, on dispose d'une fonction de hachage qui associe à chaque élément une suite (non bornée) de bits. L'index T est une table de 2^d cases et l'adresse d'une case est une suite de d bits. Chaque case de T pointe sur une page pouvant contenir au plus b éléments. Chaque page P est caractérisée par une suite $s(P)$ de k bits, avec $k \leq d$: la page P contient tous les éléments dont la valeur de hachage commence par la suite $s(P)$. Et il y a 2^{d-k} cases de T qui pointent sur une même page P : toutes les cases dont l'adresse commence par $s(P)$ (si $k = d$, il y a une seule case de T qui pointe vers P : la case d'adresse $s(P)$).

Sur la figure ci-dessous, on a considéré les éléments suivants, avec leurs valeurs de hachage (sur 6 bits) : $h(\text{tomate}) = 011101$, $h(\text{ananas}) = 001000$, $h(\text{blé}) = 001110$, $h(\text{gruau}) = 011001$, $h(\text{mangue}) = 000100$, $h(\text{pomme}) = 101110$, $h(\text{haricot}) = 101011$, $h(\text{seigle}) = 100010$, $h(\text{riz}) = 100110$, $h(\text{noix}) = 100101$, $h(\text{manioc}) = 111010$, $h(\text{abricot}) = 110111$.

On suppose que la capacité des pages est 5. On considère la situation où la taille de l'index est 4. La page P_1 contient 5 éléments dont le premier bit de la valeur de hachage est 0, donc $s(P_1) = 0$ et les 2 cases dont l'adresse commence par 0 pointent sur P_1 : il y en a 2. La page P_2 contient 5 éléments, dont la valeur de hachage commence par les bits 1 et 0, donc $s(P_2) = 10$ et la case d'adresse 10 pointe sur P_2 . La page P_3 contient 2 éléments, dont la valeur de hachage commence par les bits 1 et 1, donc $s(P_3) = 11$ et la case d'adresse 11 pointe sur P_3 .



Pour insérer un élément x , on calcule l'adresse v obtenue en prenant les d premiers bits de la valeur de hachage de x . Soit P la page pointée à cette adresse.

- Si x est dans P il n'y a rien à faire.
- Si P n'est pas pleine on y insère x .
- Si P est pleine
 - dans le cas où $v = s(P)$ il faut commencer par doubler la taille de l'index et mettre à jour les pointeurs vers les pages.
 - et dans tous les cas, on éclate P en 2 pages P_1 et P_2 telles que $s(P_1) = s(P).0$ et $s(P_2) = s(P).1$, on répartit les éléments de P dans ces 2 pages et l'on met à jour les cases de T qui pointaient sur P .
 - on insère x dans ce nouvel environnement

Question 1. Donner le résultat de l'insertion, dans la table de la figure précédente, d'un élément x_1 dont la valeur de hachage est $h(x_1)=110011$, puis d'un élément x_2 dont la valeur de hachage est $h(x_2)=011110$, et enfin d'un élément x_3 dont la valeur de hachage est $h(x_3)=101001$.

Question 2. Donner un ensemble de primitives, avec leurs spécifications, permettant de décrire les traitements de hachage extensible. (Il est conseillé de répondre à cette question en traitant la question suivante.)

Question 3. Écrire l'algorithme d'insertion d'un élément en utilisant cet ensemble de primitives.

Question 4. Expliquer les différents cas rencontrés pour la suppression d'un élément, et écrire l'algorithme de suppression en utilisant les primitives.

2 Distance minimale [10 points]

On considère un ensemble E de n points du plan. On cherche à calculer la distance minimum $dist_E$ entre 2 points de E et à retourner un couple de points à distance minimum ; la méthode proposée est dichotomique :

- diviser E en deux sous-ensembles E_g et E_d de tailles sensiblement voisines, séparés par une ligne verticale
- résoudre le problème dans chacun des deux sous-ensembles E_g et E_d
- trouver la solution pour E : $dist_E$ est soit la distance minimum dans E_g , soit la distance minimum dans E_d , soit la distance minimum entre un point de E_g et un point de E_d .

N.B. : dans ce problème on peut traiter les **questions 3.2, 4 et 5** en utilisant, sans les avoir démontrés, les résultats des questions 1, 2 et 3.1.

Un point est représenté par le couple formé de son abscisse et de son ordonnée (et on dispose de deux primitives, nommées $Pabs$ et $Pord$, prenant un point et renvoyant son abscisse –resp. son ordonnée).

On suppose que les points de E sont représentés dans deux tableaux ABS et ORD ; le premier contient tous les points dans l'ordre de leurs abscisses croissantes (et par ordonnées croissantes pour les points d'abscisses égales) ; le second contient aussi tous les points, mais rangés en ordre croissant de leurs ordonnées.

Question 1. Montrer que l'on peut, en effectuant n comparaisons, construire 4 tableaux ABS_1 , ABS_2 , ORD_1 et ORD_2 , satisfaisant les conditions suivantes :

- les tableaux ABS_1 et ABS_2 réalisent une partition de E et les nombres de points dans ABS_1 et ABS_2 différent au plus d'une unité,
- les tableaux ABS_1 et ORD_1 (respectivement ABS_2 et ORD_2) contiennent les mêmes points,
- dans ABS_1 (resp. ABS_2), les points sont rangés en ordre croissant de leurs abscisses ; et dans ORD_1 (resp. ORD_2), les points sont rangés en ordre croissant de leurs ordonnées,
- l'abscisse de tout point de ABS_1 est inférieure à l'abscisse de tout point de ABS_2 .

Remarque : tous les points de ORD_1 ont une abscisse inférieure ou égale à celle du dernier point de ABS_1 .

Question 2. (facultative) Montrer que si 5 points sont situés à l'intérieur ou sur le bord d'un carré de côté $a > 0$, alors il y a au moins 2 points dont la distance est strictement inférieure à a .

Question 3. Soit $b \in R$, on suppose que E_g et E_d est une partition de E telle que l'abscisse de tout point de E_g est inférieure ou égale à b , et l'abscisse de tout point de E_d est supérieure ou égale à b . On suppose de plus que la distance minimum $dist_{E_g}$ entre 2 points de E_g est supérieure ou égale à un certain réel a , et de même $dist_{E_d} \geq a$.

1. On appelle ORD' le tableau obtenu à partir de ORD en ne retenant que les points dont l'abscisse x vérifie $b - a \leq x \leq b + a$. Montrer que s'il existe deux points $P = (x_1, y_1)$ et $Q = (x_2, y_2)$ tels que $dist(P, Q) < a$ et $y_1 \leq y_2$, alors P et Q sont dans ORD' et Q se trouve parmi les 7 points qui suivent P dans ORD' .
2. Donner un algorithme, en $O(n)$ comparaisons, qui retourne un couple (P, Q) de points de distance minimale dans E (soit il s'agit de deux points de la même classe – E_g ou E_d –, soit la distance entre P et Q est inférieure au minimum des distances $dist_{E_g}$ et $dist_{E_d}$, et dans ce cas les deux points ne sont pas dans la même classe).

Question 4. En déduire un algorithme recherchant 2 points à distance minimale dans un ensemble E de n points.

Question 5. Donner la relation de récurrence satisfaite par le nombre de comparaisons effectuées par cet algorithme. En déduire sa complexité.

3 Arbre couvrant presque minimal [10 points]

On considère un graphe $G = (S, A)$, non orienté, connexe et contenant au moins un cycle. Chaque arête porte une valuation et **toutes les valuations sont différentes**.

Question 1. Montrer qu'il existe un **unique** arbre couvrant minimal pour G , que l'on nommera par la suite T . (On pourra admettre ce résultat sans le démontrer.)

Question 2. On appelle *arbre couvrant presque minimal* un arbre couvrant de G qui est de valuation minimale parmi tous les arbres couvrants autres que T . Donner un exemple montrant qu'il peut exister plusieurs arbres couvrants presque minimaux de G .

Question 3. Montrer qu'il existe un arbre couvrant presque minimal obtenu à partir de T en échangeant une arête de T et une arête de G (c'est-à-dire qu'il existe une arête $a_1 \in T$ et une arête $a_2 \notin T$ telles que $T - a_1 + a_2$ est arbre couvrant presque minimal).

Question 4. Étant donnés deux sommets u et v de G , on note $\max(u, v)$ une arête de valuation maximale sur l'unique chemin de u à v dans T . Décrire un algorithme permettant de calculer $\max(u, v)$, pour tout couple de sommets de G , en temps $O(n^2)$ où n est le nombre de sommets de G .

Question 5. En supposant que l'arbre couvrant minimal T a déjà été calculé, donner un algorithme qui calcule un arbre couvrant presque minimal de G en temps $O(p + n^2)$ où p est le nombre d'arêtes de G .