TD 12 : Algorithmes probabilistes

1 Tables de hachage

Un dictionnaire est un annuaire dont la fréquence de modifications est très faible devant la fréquence des consultations. Par exemple, les correcteurs orthographiques des traitements de texte ont recours à un dictionnaire.

Dans la suite, on s'intéresse a des dictionnaires implementés par des tables de hachage. Dans la mesure où les données du dictionnaire sont pratiquement inchangées durant une longue période, il est intéressant de choisir la fonction de hachage parmi un ensemble de fonctions de telle sorte qu'il n'y a pas de collisions.

Supposons par la suite que les identifiants sont n valeurs numériques comprises entre 0 et $p-1$, avec p premier. Nous désirons stocker ce dictionnaire dans une table de hachage à m entrées, avec $m \leq p$.

La fonction de hachage que nous recherchons se trouve parmi l'ensemble $H = \{h_{a,b} \mid 0 < a < p \wedge 0 \leq b < p\}$ où

$$h_{a,b}(x) = ((ax + b \mod p) \mod m) + 1$$

Exercice 1.

1. *Soient i et j deux identifiants différents et $h_{a,b} \in H$. Montrer que $ai + b \not\equiv aj + b \mod p$.*
2. *Soient i et j deux identifiants différents. Montrer que*

   $$\forall 0 \leq u \neq v < p,\ \exists! h_{a,b} \in H \text{ t.q. } (ai+b \equiv u \mod p) \wedge (aj+b \equiv v \mod p)$$

3. *On suppose toujours $i \neq j$. Montrer que $Card\{h_{a,b} \mid h_{a,b}(i) = h_{a,b}(j)\} \leq \frac{p(p-1)}{m}$*
4. *En déduire que $q = \sum_{i<i'} Card\{(a,b) \mid h_{a,b}(i) = h_{a,b}(i')\}$ le nombre total des collisions vérifie $q \leq \frac{n(n-1)}{2}\frac{p(p-1)}{m}$.*
5. *Soit $m = n^2$. Montrer que $Card(H') < \frac{1}{2}Card(H)$, où H' est le sous-ensemble de fonctions qui produisent au moins une collision.*
6. *Donner un algorithme probabiliste pour trouver une fonction de hachage sans collisions. Analyser l'algorithme.*

Exercice 2. *La table de hachage dans l'exercice précèdent a n^2 entrées. Nous allons obtenir un meilleur résultat avec un* double *hachage.*

Le double hachage opère de la façon suivante:

- *Chaque entrée de la table de hachage* primaire *est une référence vers une table de hachage* secondaire*.*
- *Chaque table de hachage secondaire possède sa fonction de hachage propre. On note h_i la fonction de hachage associée à l'entrée i de la table primaire et m_i la taille de la table de l'entrée i.*
- *Lorsqu'un nouvel identifiant id est ajouté à la table on calcule $i = h(id)$ puis $j = h_i(id)$ afin de déterminer dans quelle entrée de la table de hachage référencée depuis l'entrée i, il doit être inséré. L'insertion se fait comme pour un table de hachage simple.*
- *La recherche et la suppression suivent un schéma similaire.*

1. *Soit $H'' \subseteq H$ le sous-ensemble des fonctions de hachage qui produisent au moins n collisions. Montrer que, lorsque $m = n$, $Card(H'') \leq \frac{1}{2} Card(H)$.*

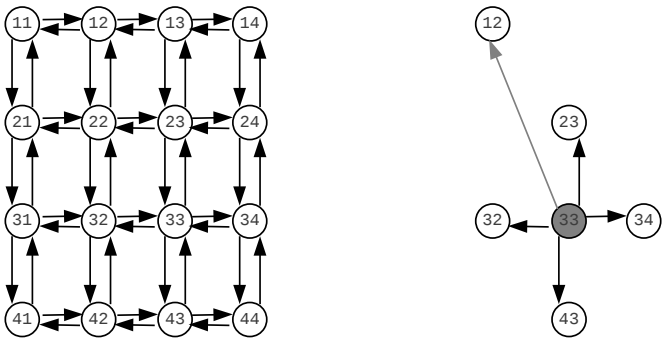
2. *On choisit une fonction $h \notin H''$ comme fonction de hachage primaire. Notons n_i le nombre d'identifiants id du dictionnaire t.q. $h(id) = i$. En déduire que $\sum_i \frac{n_i(n_i-1)}{2} < n$.*

3. *Comment peut-on choisir les fonctions de hachage secondaires? Quelle est la complexité en espace de ce double hachage?*

2 Routage randomisé

Nous considérons un graphe aléatoire orienté (représentant un réseau social) construit de la façon suivante :

- L'ensemble des sommets V est défini par $V = \{(i, j) \mid 1 \leq i, j \leq n\}$ où n est un paramètre du modèle. La *distance* entre deux sommets est définie par $d((i, j), (i', j')) = |i' - i| + |j - j'|$.
- Les arcs *fixes* forment une grille : il y a un arc entre toute paire de sommets à distance 1 (voir la partie gauche de la figure ci-dessous). Ces sommets sont dits *voisins*.
- Tout sommet $u = (i, j)$ a de plus un arc *variable* vers un sommet aléatoire (voir la partie droite de la figure ci-dessous). La *longueur* d'un tel arc est la distance entre ses deux extrémités. La distribution de ce sommet $pl(u)$ sera précisée ultérieurement. Ce sommet aléatoire peut être u ou l'un de ses voisins sur la grille et dans ce cas l'arc n'est pas ajouté.



Etant donnés deux sommets s, t et un message à envoyer de s à t, on étudie l'algorithme qui consiste à chaque étape à envoyer par un arc du graphe le message au sommet le plus proche (au sens de la distance d) de t. Par exemple dans le cas de la figure ci-dessus et d'un message à envoyer de $(3,3)$ à $(1,1)$, le sommet $(3,3)$ l'envoie au sommet $(1,2)$ qui l'envoie à son tour au sommet $(1,1)$. L'objet du problème est d'étudier l'espérance du nombre d'étapes, noté N_e, que devra parcourir le message selon la nature de la distribution pl.

Exercice 3. *Dans cette partie, pour tout u, v, $pl\,(u,v) = n^{-2}$. Autrement dit, on a affaire à une distribution uniforme.*

1. *Soit W l'ensemble des sommets à distance au plus $n^{\frac{2}{3}}$ de t. Démontrer que $|W| \leq 4n^{\frac{4}{3}}$ pour $n \geq 4$.*

2. *Démontrer que la probabilité que la destination d'un arc variable appartienne à W est inférieure ou égale $4n^{-\frac{2}{3}}$.*

   Pour les deux questions suivantes on suppose que Δ, la distance de s à t, vérifie $\Delta > n^{\frac{2}{3}}$.

3. *Démontrer que $\mathbf{P}(N_e < \frac{1}{8}n^{\frac{2}{3}}) \leq \frac{1}{2}$.*

4. *En déduire que $\mathbf{E}(N_e) \geq \frac{1}{16}n^{\frac{2}{3}}$.*

Exercice 4. *Dans cette partie, pour tout $u \neq v$, $pl\,(u,v) = d(u,v)^{-3} / \sum_{v' \in V \setminus \{u\}} d(u,v')^{-3}$ et $pl(u,u) = 0$. Autrement dit, la distribution est inversement proportionnelle au cube de la distance.*

1. *Soit v le sommet destination de l'arc variable issu d'un sommet u. Démontrer que $\mathbf{P}(d(u,v) > m) \leq \frac{4}{m}$ (avec $m \geq 1$).*

2. *Soit V' un sous-ensemble de sommets tel que $|V'| \leq \frac{\sqrt{n}}{8}$. Démontrer que la probabilité qu'au moins un arc variable issu de V' soit de longueur supérieure à $\sqrt{n}$ est inférieure ou égale à $\frac{1}{2}$.*

3. *Pour les deux questions suivantes on suppose que* Δ*, la distance de* s *à* t*, vérifie* $\Delta > \frac{n}{8}$.

   Démontrer que $\mathbf{P}(N_e \leq \frac{\sqrt{n}}{8}) \leq \frac{1}{2}$.

4. *En déduire que* $\mathbf{E}(N_e) \geq \frac{\sqrt{n}}{16}$.

Exercice 5. *Dans cette partie, pour tout* $u \neq v$, $pl\,(u,v) = d(u,v)^{-2} / \sum_{v' \in V \setminus \{u\}} d(u,v')^{-2}$ *et* $pl(u,u) = 0$. *Autrement dit, la distribution est inversement proportionnelle au carré de la distance. "*ln*" désigne le logarithme népérien et "*log*" le logarithme en base 2.*

1. *Démontrer que pour tout* u, $\sum_{v' \in V \setminus \{u\}} d(u,v')^{-2} \leq 4\ln(6n)$.

   On dit que l'algorithme est dans la phase k*, si* dst*, la distance du sommet courant* u *à la destination* t*, vérifie* $2^{k-1} < dst \leq 2^k$. *La phase* -1 *correspond à l'arrivée du message.*

2. *Supposons que l'algorithme soit dans la phase* k *(avec* $2^{k-1} < 2n-2$*). Démontrer que la probabilité qu'il soit à l'étape suivante dans une phase inférieure à* k *est supérieure ou égale à* $\frac{1}{256\ln(6n)}$.

3. *En déduire que le nombre moyen d'étapes pour passer d'une phase* k *à une plus petite phase est inférieur ou égal à* $256\ln(6n)$.

4. *En déduire que* $\mathbf{E}(N_e) = O(\log^2(n))$.