TD 3 : Analyse amortie, analyse en moyenne et analyse probabiliste

Exercice 1. *File sur piles*

Grâce aux tableaux dynamiques, on peut implanter une structure de données de pile dont les deux opérations (EMPILER et DÉPILER) ont une complexité amortie constante. On souhaite implanter une file.

1. Pourquoi les tableaux dynamiques ne suffisent pas *a priori* ?

Pour implanter la file, on utilise deux piles E (entrée) et S (sortie). Pour ENFILER un élément, on l'empile sur E. Pour défiler, on dépile depuis S. Si S est vide et qu'on doit DÉFILER, on commence par transférer tous les éléments de E dans S.

2. Écrire les deux algorithmes ENFILER et DÉFILER et analyser leurs complexités dans le pire des cas. *On suppose avoir accès aux opérations de base sur les piles.*

3. On va démontrer que le coût amorti par opérations est constant, avec les différentes méthodes.

- **i.** Utiliser la *méthode de l'agrégat* : montrer que si on part d'une file vide et qu'on effectue n opérations ENFILER/DÉFILER, il y aura au plus $3n$ opérations EMPILER/DÉPILER effectuées.
- **ii.** Utiliser la *méthode des acomptes* : au moment d'ENFILER un élément x, on lui fournit un acompte de 2. Montrer qu'avec cet acompte, chaque opération a un coût amorti constant.
- **iii.** Utiliser la *méthode du potentiel*, en définissant le potentiel de la file (E,S) comme étant deux fois la taille de E.

Exercice 2. *Compteur avec incrément et décrément*

On cherche à maintenir un compteur qui peut alterner les incréments et les décréments. *La mesure de coût reste la même que dans le cours : chaque inversion de la valeur d'un bit coûte* 1.

1. Montrer qu'avec un compteur standard, il est possible que le coût amorti ne soit pas constant.

Pour retrouver un coût *amorti* constant par opération, on représente la valeur v du compteur par un couple de deux entiers (P,N) tel que $v = P - N$. Parmi les représentations de v, on appelle *représentation exclusive* un couple (P,N) qui vérifie $P \wedge N = 0$, où $\wedge$ représente un ET logique, bit-à-bit. *Par exemple,* $18 \wedge 9 = 0$ *car* $18 = 10010_2$ *et* $9 = 01001_2$*, donc* $(18,9)$ *est une représentation exclusive de la valeur* 7*. Par contre,* $20 \wedge 13 = 4$ *donc* $(20,13)$ *n'est pas une représentation exclusive de* 7*.*

2. Soit (P,N) un couple d'entiers quelconque et $v = P - N$.

- **i.** Supposons que les $k^{\text{èmes}}$ bits $P_{[k]}$ et $N_{[k]}$ de P et N sont tous deux à 1. Montrer qu'en les passant à 0, on obtient un nouveau couple (P',N') tel que $P' - N' = v$.
- **ii.** En déduire que tout entier v possède une représentation exclusive (P,N).
- **iii.** Montrer que tout entier non nul v possède une infinité de représentations exclusives (P,N).
- **iv.** Que peut-on dire pour l'entier $v = 0$?

L'idée du compteur (P,N) est d'incrémenter P pour incrémenter le compteur, et incrémenter N pour le décrémenter, tout en conservant l'invariant $P \wedge N = 0$.

3.
- **i.** Soit (P,N) un représentation exclusive de v, et $P_{++} = P + 1$. Expliquer comment obtenir, à partir du couple (P_{++},N), une représentation exclusive (P',N') de $v+1$.
- **ii.** Combien d'inversions de bits sont nécessaires pour passer de (P_{++},N) à (P',N'). ?
- **iii.** Écrire les deux algorithmes INCRÉMENT et DÉCRÉMENT qui prennent en entrée une représentation exclusive (P,N) d'un entier v, et renvoient une représentation exclusive (P',N') de $v+1$ et $v-1$, respectivement.

4. Quel est le coût individuel maximal d'une opération d'incrément ou de décrément ?

On cherche maintenant à faire une analyse *amortie* de ce compteur, avec la méthode du potentiel. Soit $v_0 = 0$ la valeur initiale du compteur, représentée par le couple $(P_0,N_0) = (0,0)$. On effectue t opérations INCRÉMENT ou DÉCRÉMENT successives. On note v_i la iˆème valeur prise par le compteur, et (P_i,N_i) le couple correspondant. On note c_i le nombre de bits inversés lors de la $i^{\text{ème}}$ opération et on définit le potentiel Φ_i comme étant le nombre total de bits à 1 dans (P_i,N_i).

5. **i.** Exprimer le *coût amorti* a_i de la $i^{\text{ème}}$ opération, en fonction de c_i, Φ_i et Φ_{i-1}.
   ii. Montrer que $a_i \leq 2$.
   iii. Conclure.
6. On suppose maintenant que l'état initial du compteur est une valeur v représentée par $(v, 0)$ (si $v > 0$) ou $(0, v)$ (dans le cas inverse). Que dire du coût amorti par opération ?

Exercice 3. *Algorithme d'Euclide*

L'algorithme d'Euclide permet de calculer le PGCD de deux entiers, en faisant des divisions euclidiennes successives. On note $a \bmod b$ le reste dans la division euclidienne de a par b, qui vérifie $0 \leq a \bmod b < b$. *Dans la description de l'algorithme ci-dessous, on suppose sans perte de généralité que $a > b \geq 0$.*

EUCLIDE(a, b) :

```
1 Tant que b > 0 :
2     (a, b) ← (b, a mod b)
3 Renvoyer a
```

1. Montrer que l'algorithme termine et que le nombre d'itérations est $\leq b$.

On veut utiliser la technique de l'analyse amortie pour donner une borne bien meilleure sur le nombre d'itérations. La valeur de b ne décroît pas suffisamment à chaque étape : à certaines étapes, elle ne diminue que de 1. Cependant, on définit une *fonction potentielle* qui décroît beaucoup plus régulièrement.
Pour cela, on note a_i et b_i les valeurs de a et b après la $i^{\text{ème}}$ itération, en posant $a_0 = a$ et $b_0 = b$ initialement. Et on définit $s_i = a_i + \frac{1}{\varphi} b_i$ où $\varphi = \frac{1}{2}(1 + \sqrt{5}) \simeq 1,618$ est le nombre d'or, qui vérifie $\varphi^2 - \varphi - 1 = 0$.

2. Exprimer a_{i+1} et b_{i+1} en fonction de a_i et b_i pour $i \geq 0$.
3. Montrer que $b_{i+1} \leq a_i - b_i$. *Utiliser le fait que le quotient dans la division euclidienne de a_i par b_i est ≥ 1.*
4. En déduire que $s_{i+1} \leq \frac{1}{\varphi} s_i$. *Montrer pour cela que $\varphi - 1 = \frac{1}{\varphi}$.*
5. En déduire que $s_i \leq \frac{1}{\varphi^i} s_0$ pour tout $i \geq 0$.
6. Soit k le nombre d'itérations effectuées par l'algorithme.
   i. Montrer que $s_{k-1} \geq 2 + \frac{1}{\varphi} = \varphi^2$.
   ii. En déduire que $k \leq 3 + \log_\varphi(a + \frac{1}{\varphi} b)$.
7. En déduire que $k = O(\log b)$. *Indication : ce n'est pas totalement immédiat, mais presque.*

Exercice 4. *Recherche de motif*

On s'intéresse à la recherche de motif dans un texte. En entrée, on dispose d'un *texte* t de longueur n et d'un motif x de longueur m, avec $m < n$. On cherche s'il existe une occurence de x dans t, c'est-à-dire s'il existe un indice i tel que $t_{[i,i+m[} = x$, où $t_{[i,i+m[}$ est le sous-mot constitué des lettres $t_{[i]}$, $t_{[i+1]}$, …, $t_{[i+m-1]}$. Le sous-mot $t_{[i,i+m[}$ est appelé *fenêtre de longueur m en position i*.

1. Écrire l'algorithme *naïf* de recherche de motif dans un texte, et analyser sa complexité en pire cas.

On cherche maintenant à effectuer une analyse *en moyenne* de l'algorithme précédent. On suppose dans un premier temps que le texte est fixé, mais le motif tiré aléatoirement et uniformément : chaque lettre x_j est tirée de manière uniforme et indépendante dans un alphabet Σ.

2. On fixe une fenêtre de longueur $t_{[i,i+m[}$.
   i. Quelle est la probabilité que $x = t_{[i,i+m[}$.
   ii. Quelle est l'espérance du nombre de lettres communes au début de $t_{[i,i+m[}$ et de x ? *On demande l'espérance de j tel que $t_{[i,i+j[} = x_{[0,j[}$ et $t_{[i,i+j+1[} \neq x_{[0,j+1[}$.*
   iii. En déduire l'espérance du nombre de comparaisons effectuées durant l'algorithme.

Exercice 5. QUICKSELECT *particulier*

1. On s'intéresse à l'espérance E_n du nombre de comparaisons effectuées par QUICKSELECT$(T, 1)$ où T est un tableau de n éléments.
   i. Montrer que $E_n = (n-1) + \frac{1}{n} \sum_{m=1}^{n-1} E_m$.
   ii. En déduire que $E_n \leq 2n - 2$.

2. De même, étudier l'espérance du nombre de comparaisons effectuées par QUICKSELECT$(T, 2)$.
3. Donner des algorithmes déterministes (simples !) pour ces deux cas, et comparer les complexités obtenues.

Exercice 6. *Compteur probabiliste*

Un compteur entier qui va de 0 à n utilise un espace mémoire $O(\log n)$. On va voir une technique probabiliste qui permet de fournir un compteur approximatif qui utilise un espace exponentiellement plus petit. L'idée est d'avoir une fonction INCRÉMENT qui soit probabiliste.

1. On définit une fonction INCRÉMENT(c) qui incrémente c avec probabilité p et qui ne fait rien avec probabilité $1-p$. On initialise c à 0 et on effectue n appels à INCRÉMENT.

i. Quelle est l'espérance de la valeur de c ?
ii. Quelle valeur faut-il renvoyer pour avoir une valeur approchée de n ?
iii. Borner la probabilité que la valeur renvoyée soit $\geq 2n$.

On modifie la fonction INCRÉMENT, pour que INCRÉMENT(c) incrémente c avec probabilité $1/2^c$ et ne fasse rien avec probabilité $1-1/2^c$.

2. **i.** On suppose avoir accès à une fonction BITALÉATOIRE() qui renvoie 0 avec probabilité $\frac{1}{2}$ et 1 avec probabilité $\frac{1}{2}$. Écrire explicitement la fonction INCRÉMENT.
ii. Quelle est l'espérance du nombre d'appels effectués à BITALÉATOIRE ?

On note C_k la variable aléatoire qui décrit la valeur du compteur après k appels à INCRÉMENT, et $p_{k,c} = \Pr[C_k = c]$ la probabilité que le compteur ait la valeur c après k appels à INCRÉMENT.

3. **i.** Calculer $p_{0,0}$, $p_{0,c}$ pour $c > 0$ et $p_{k,0}$ pour $k > 0$.
ii. Calculer $\sum_{c\geq 0} p_{k,c}$.
iii. Montrer que pour $k, c > 0$, $p_{k,c} = \frac{1}{2^{c-1}} p_{k-1,c-1} + (1-\frac{1}{2^c}) p_{k-1,c}$. *Indication. Si le compteur vaut c après k appels à* INCRÉMENT*, quelle peut être sa valeur après $k-1$ appels ?*

Finalement, on décide de renvoyer la valeur $v = 2^c$. On cherche à calculer l'espérance de la valeur finale de v après n appels à INCRÉMENT. On note $V_k = 2^{C_k}$ la variable aléatoire qui décrit la valeur v renvoyée par l'algorithme.

4. **i.** Exprimer $\mathbb{E}[V_k]$ en fonction de $p_{k,c}$, pour $k, c \geq 0$.
ii. En déduire que $\mathbb{E}[V_k] = \mathbb{E}[V_{k-1}] + 1$.
iii. En déduire que $\mathbb{E}[V_n] = n$.

Exercice 7. *Méthode de Monte Carlo*

La méthode de Monte Carlo permet de calculer une valeur approchée de π, et de manière plus générale d'estimer diverses valeurs numériques. Pour calculer π, l'idée est d'estimer l'aire d'un quart de cercle de rayon 1.

1. Quelle est l'aire d'un quart de cercle de rayon 1 ?

Afin d'estimer l'aire du quart de cercle, on tire aléatoirement des points dans un carré de côté 1, et on compte la proportion qui se situe dans le quart de cercle. On peut montrer [1] que l'espérance de cette proportion est l'aire du quart de cercle. On suppose avoir accès à une fonction RANDOM() qui renvoie un réel tiré uniformément entre 0 et 1.

2. Écrire une fonction MONTECARLO(n) qui estime π à l'aide de cette méthode, en tirant n points aléatoires. *Essayer de ne pas utiliser de fonction « racine carrée ».*
3. La méthode de Monte-Carlo est bien plus générale. Soit par exemple f une fonction de $[0,1]$ dans $[0,1]$. Proposer un généralisation de l'algorithme précédent qui prenne en entrée f et n et calcule une approximation de $\int_0^1 f(x)dx$.

1. Cela nécessite formellement de travailler avec des probabilités continues, donc on omet cette démonstration.