

Cache de pages Web

Épreuve pratique d'algorithmique et de programmation
Concours commun des écoles normales supérieures

Durée de l'épreuve: 3 heures 30 minutes

Juin/Juillet 2011

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Sur votre table est indiqué un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un \tilde{u}_0 particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion!). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec \tilde{u}_0 au lieu de u_0 . Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examineur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures!

Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n , on demande l'ordre de grandeur en fonction du paramètre, par exemple: $O(n^2)$, $O(n \log n)$,...

Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de **tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe**. Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

1 Introduction

On s'intéresse à un serveur dont le rôle est de télécharger des pages Web et de les fournir à un utilisateur. Le serveur possède une mémoire propre, qu'on appelle cache, qui lui sert à stocker des pages Web afin de limiter le nombre de téléchargements. Ce cache est de taille limitée, et peut stocker jusqu'à k pages Web (on suppose que toutes les pages ont la même taille). Lorsque l'utilisateur soumet une requête pour une certaine page Web, le serveur doit d'abord charger la page dans son cache si elle ne s'y trouve pas déjà, puis il la transmet à l'utilisateur. Lors d'un chargement, si le cache est plein, il doit décider quelle page doit être évincée du cache. Si le cache n'est pas plein, ou que la page demandée est dans le cache, il n'est pas nécessaire d'éliminer une page du cache.

On ne considère pas ici les problèmes de mises à jour des pages : lorsqu'une page est dans le cache, elle peut être fournie directement à l'utilisateur.

On se propose d'étudier les stratégies de gestion du cache, dans le but de minimiser le nombre de chargements de pages, pour des séquences de requêtes soumises par un utilisateur. On considère qu'il existe P pages possibles, référencées par leur numéro, entre 0 et $P - 1$. Sauf mention contraire, on considère que le cache est vide avant la première requête.

2 Génération de séquences de requêtes

Considérons la suite d'entiers (u_k) définie pour $k \geq 0$ par :

$$u_k = \begin{cases} \text{votre } u_0 \text{ (à reporter sur votre fiche)} & \text{si } k = 0 \\ 15\,091 \times u_{k-1} \pmod{64\,007} & \text{si } k \geq 1 \end{cases}$$

Question 1 Que valent : **a)** u_{10} **b)** u_{100} **c)** u_{1000}

On considère la séquence $S_{n,P}$ de n pages $S_{n,P}(0), \dots, S_{n,P}(n-1)$ définie par

$$S_{n,P}(k) = \left\lfloor 2 \times P \times \left(1 - \frac{1000}{(u_k \pmod{1000}) + 1000} \right) \right\rfloor$$

Question 2 Donnez les séquences suivantes : **a)** $S_{4,10}$ **b)** $S_{4,20}$ **c)** $S_{4,30}$

Question 3 Combien de chargements sont nécessaires pour traiter les séquences suivantes avec un cache de taille un (une seule page peut être stockée) ?

a) $S_{5,5}$ **b)** $S_{50,5}$ **c)** $S_{500,5}$

3 Stratégies classiques

On considère la stratégie FIFO (de l'anglais *First In, First Out*) : lorsqu'une page doit être éliminée du cache, on choisit toujours celle qui a été chargée la première, parmi toutes les pages du cache.

Question 4 *Donnez le nombre de chargements nécessaires à la stratégie FIFO pour traiter les séquences suivantes avec un cache de taille k :*

- a) $S_{10,6}, k = 3$ b) $S_{50,20}, k = 10$ c) $S_{500,50}, k = 10$ d) $S_{5000,50}, k = 10$

Pour prendre en compte le fait que certaines pages reviennent plus souvent que d'autres, on définit la stratégie LRU (de l'anglais *Least Recently Used*) : dans cette politique, lorsqu'une page doit être évincée du cache, on choisit celle dont la dernière requête est la plus ancienne.

Question 5 *Donnez le nombre de chargements nécessaires à la stratégie LRU pour traiter les séquences suivantes avec un cache de taille k :*

- a) $S_{10,6}, k = 3$ b) $S_{50,20}, k = 10$ c) $S_{500,50}, k = 10$ d) $S_{5000,50}, k = 10$

Question à développer pendant l'oral : Donner la complexité de votre algorithme.

On définit enfin la stratégie LFU (de l'anglais *Least Frequently Used*), qui conserve, pour toutes les pages du caches, le nombre de requêtes à ces pages. Lorsqu'une page doit être éliminée du cache, on choisit celle dont le nombre de requêtes est le plus faible. En cas d'égalité, on utilise la politique LRU pour choisir la page à éliminer parmi celles dont le nombre de requêtes est minimal.

Question 6 *Donnez le nombre de chargements nécessaires à la stratégie LFU pour traiter les séquences suivantes avec un cache de taille k :*

- a) $S_{10,6}, k = 3$ b) $S_{50,20}, k = 10$ c) $S_{500,50}, k = 10$ d) $S_{5000,50}, k = 10$

Les parties 4, 5 et 6 sont indépendantes, et de difficulté croissante.

4 Stratégie optimale

Dans cette partie, on considère que l'on connaît à l'avance toute la séquence de requêtes de pages. Bien sûr, cette hypothèse est irréaliste en pratique, mais elle permet de concevoir un algorithme optimal, et de comparer les autres stratégies par rapport à cet optimal.

On peut montrer que si l'on connaît la séquence des requêtes, et qu'une page doit être évincée du cache, la stratégie optimale est de choisir celle qui réapparaît le plus loin possible dans la séquence : ce peut être soit une page qui ne réapparaît plus du tout, soit, si toutes les pages en cache réapparaissent dans la séquence, celle dont la prochaine occurrence est la plus tardive.

Question 7 *En vous appuyant sur l'observation précédente, calculez le nombre minimal de chargements de pages nécessaires pour traiter les séquences suivantes, avec une taille de cache k :*

- a) $S_{10,6}, k = 3$ b) $S_{50,20}, k = 10$ c) $S_{500,50}, k = 10$ d) $S_{5000,50}, k = 10$

On considère un algorithme A quelconque pour le problème du cache (qui ne connaît pas à l'avance la séquence de requêtes), et l'algorithme optimal ci-dessus, noté OPT. On suppose que A dispose d'un cache de taille k_A et OPT d'un cache de taille k_{OPT} , avec $k_{OPT} \leq k_A$. Initialement, les caches sont pleins : l'ensemble des pages initialement dans le cache de A est noté S_A^{init} alors que l'ensemble des pages dans le cache de OPT est noté S_{OPT}^{init} .

On considère la séquence de pages décrit par ces deux étapes :

1. Les $k_A - k_{OPT} + 1$ premières pages de la séquence ne sont ni dans S_A^{init} ni dans S_{OPT}^{init} . On note S_1 l'ensemble de ces pages.
2. Les $k_{OPT} - 1$ pages suivantes sont définies de la façon suivante : à chaque instant, on choisit une page de $S_{OPT}^{init} \cup S_1$ qui n'est pas dans le cache de A.

Questions à développer pendant l'oral :

- (a) Justifiez que pour la deuxième étape, on peut toujours trouver une page de $S_{OPT}^{init} \cup S_1$ qui n'est pas dans le cache de A.
- (b) Calculez le nombre de chargements de pages pour l'algorithme A, ainsi que pour OPT.
- (c) En déduire une borne sur l'efficacité d'un algorithme ne connaissant pas à l'avance la séquence de pages.

5 Réordonnancement partiel

Dans cette partie, on suppose que l'on connaît les r prochaines requêtes, à partir de la première requête non traitée, et que l'on peut choisir de servir n'importe laquelle de ces requêtes. Formellement, pour une séquence de pages p_0, \dots, p_{n-1} , si la première requête non traitée est p_i , on peut traiter toute requête dans la fenêtre $[p_i, p_{i+1}, \dots, p_{i+r-1}]$. Une requête traitée est marquée comme telle, mais n'est pas supprimée de la fenêtre.

On considère la stratégie suivante : tant qu'il est possible de traiter des requêtes dans la fenêtre sans chargement de pages, on les traite (et on déplace la fenêtre le plus possible vers la droite). Quand il n'est plus possible de traiter de requête sans chargement, on traite la première requête de la fenêtre (et donc on fait un chargement). Lorsqu'une page doit être éliminée du cache, on utilise la stratégie FIFO décrite à la partie 3.

Question 8 *Donnez le nombre de chargements nécessaires pour traiter les séquences suivantes à l'aide de cette stratégie, avec un cache de taille k et une taille de fenêtre r :*

- a)** $S_{10,6}, k = 3, r = 2$ **b)** $S_{50,20}, k = 10, r = 5$ **c)** $S_{500,50}, k = 10, r = 10$
d) $S_{5000,50}, k = 10, r = 10$

On cherche à calculer le nombre minimal de chargements nécessaires pour une séquence que l'on connaît complètement à l'avance. On suppose ici que le cache peut contenir une seule page ($k = 1$). On appelle q_1, q_2, \dots, q_m les premières instances des pages de la fenêtre qui ne sont pas dans le cache.

Questions à développer pendant l'oral : Justifiez que, dans l'algorithme optimal, si on note $[p_i, p_{i+1}, \dots, p_{i+r-1}]$ la fenêtre à un instant donné, alors

- (a) $q_1 = p_i$,
- (b) lorsque les pages de la fenêtre qui sont déjà présentes dans le cache ont été servies, on peut se restreindre à traiter soit q_1 , soit q_2 .

On considère une fenêtre de taille $r = 2$. On note (i, b, x) l'état du système, défini de la façon suivante :

- i est l'indice dans la séquence p_0, \dots, p_{n-1} de la première requête non traitée (donc la fenêtre est $[p_i, p_{i+1}]$);
- $b = 1$ si p_{i+1} a déjà été traitée, et 0 sinon;
- x est la page présente dans le cache.

On note $C(i, b, x)$ le nombre minimal de chargements pour atteindre l'état (i, b, x) . Si cet état n'est pas accessible, on note $C(i, b, x) = \perp$.

Question à développer pendant l'oral : Donnez les états accessibles à partir de (i, b, x) , ainsi que le nombre minimal de chargements nécessaires pour les atteindre.

NB : La programmation nécessaire pour la question suivante est particulièrement délicate. On conseille aux candidats d'aborder d'abord la question 10, avant de revenir sur cette question.

Question 9 Calculez le nombre minimal de chargements nécessaires pour traiter les séquences suivantes avec un cache de taille $k = 1$ et une fenêtre de taille $r = 2$:

- a) $S_{10,6}$ b) $S_{50,20}$ c) $S_{500,50}$ d) $S_{5000,50}$

Question à développer pendant l'oral : Quelle est la complexité de votre algorithme? Comment peut-on l'améliorer? Comment pourrait-on l'adapter pour prendre en compte une fenêtre de taille quelconque?

6 Coûts de chargement variables

Dans cette partie, on considère que les pages à charger proviennent de sources diverses, et que certaines sont plus coûteuses que d'autres à charger. Chaque page p ($p \in \{0, \dots, P - 1\}$) est donc munie d'un coût $c(p)$, qui correspond à un prix à payer lors du chargement de la page, et défini par :

$$c(p) = 1 + \left\lfloor \frac{400 \times u_p}{64007} \right\rfloor$$

Le coût de traitement d'une séquence de requêtes est donc la somme des coûts de pages qui doivent être chargées dans le cache, pour une politique donnée.

Question à développer pendant l'oral : On considère un cache de taille 2, et une séquence de pages (p_0, \dots, p_{n-1}) . Pour $x \neq p_i$, on note $C(i, x)$ le coût minimum pour atteindre la page p_i dans la séquence de pages avec p_i et x dans le cache. Pour simplifier, on pourra considérer que le cache contient toujours 2 pages, et qu'il y a nécessairement deux chargements initiaux. Donner une expression de $C(i, x)$ en fonction des $C(i - 1, y)$ (pour $i > 0$) ainsi que $C(0, x)$. On prendra soin de distinguer tous les cas, en fonction des valeurs de p_i, p_{i-1} et x .

Question 10 En déduire le coût optimal pour traiter les séquences suivantes, avec un cache de taille 2 :

- a) $S_{10,6}$ b) $S_{50,20}$ c) $S_{500,50}$

Question à développer pendant l'oral : Quelle est la complexité de votre algorithme? Comment l'adapter pour connaître la stratégie optimale (l'ordre des pages évincées)?

Fiche réponse type: Cache de pages Web

\widetilde{u}_0 : 12

Question 1

a)

b)

c)

Question 2

a)

b)

c)

Question 3

a)

b)

c)

Question 4

a)

b)

c)

d)

Question 5

a)

b)

c)

d)

Question 6

a)

b)

c)

d)

Question 7

a)

b)

c)

d)

Question 8

a)

b)

c)

d)

Question 9

a)

b)

c)

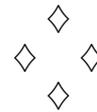
d)

Question 10

a)

b)

c)



Fiche réponse: Cache de pages Web

Nom, prénom, u₀:

Question 1

- a)
- b)
- c)

Question 2

- a)
- b)
- c)

Question 3

- a)
- b)
- c)

Question 4

- a)
- b)
- c)
- d)

Question 5

a)

b)

c)

d)

Question 6

a)

b)

c)

d)

Question 7

a)

b)

c)

d)

Question 8

a)

b)

c)

d)

Question 9

a)

b)

c)

d)

Question 10

a)

b)

c)

