MPRI2

Cours 2-15 : Analyse d'algorithmes

— 1er décembre 2005 —-

Cette épreuve est à faire en 3 heures en tout. Il est demandé de mettre en relief les idées plutôt que les détails de calcul. La notation tiendra compte de la qualité de présentation des arguments. **You can write your exam paper in English !**

Les notes de cours et autres documents peuvent être utilisés. Bonne chance !

Problème I

Permutations et tableaux

Ce problème traite de la permutation d'éléments de tableaux “en place”, c'est-à-dire avec très peu de mémoire auxiliaire. L'algorithme analysé sert dans le cas où les éléments de tableau sont de très gros enregistrements. L'analyse se transpose ensuite combinatoirement à un algorithme simple de recherche de maximum.

Rappels. Une *permutation* σ de taille $|\sigma| = n$ est un bijection de l'intervalle entier $[1\,..\,n]$ sur lui-même décrite par sa table :

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix}.$$

Le *graphe* Graphe(σ) de la permutation σ est obtenu en prenant $[1\,..\,n]$ comme ensemble de sommets et en plaçant une flèche orientée (arête) de i vers $\sigma(i)$.

Une permutation circulaire de l'ensemble $\{i_1, \ldots, i_l\}$ est de la forme

$$\tau = \begin{pmatrix} i_1 & i_2 & \cdots & i_{\ell-1} & i_\ell \\ i_2 & i_3 & \cdots & i_\ell & i_1 \end{pmatrix} \quad \text{aussi écrite} \quad \tau = [i_1, i_2, \ldots, i_\ell].$$

La *décomposition en cycles* d'une permutation peut être prise sous la forme

(DECOMP) $$\sigma = \tau_1 \circ \tau_2 \circ \cdots \circ \tau_r.$$

Ici chaque τ_j est une permutation circulaire d'un certain sous-ensemble E_j de $[1\,..\,n]$, la famille E_j formant une partition de $[1\,..\,n]$. (La notation “$\circ$” représente la composition usuelle de fonctions.)

On note $\mathcal{P}_n$ la famille de toutes les permutations de $[1\,..\,n]$ et $\mathcal{K}_n$ la sous-famille des permutations circulaires. On note $\mathcal{P} = \bigcup_n \mathcal{P}_n$ et $\mathcal{K} = \bigcup_n \mathcal{K}_n$

Q1. Justifier brièvement (par exemple, en s'aidant de Graphe(σ)), l'égalité combinatoire valable dans l'univers étiqueté :

$$\mathcal{P} = \textsc{Set}(\mathcal{K}), \qquad \mathcal{K} = \textsc{Cyc}(\mathcal{Z}).$$

Écrire les fonctions génératrices exponentielles (EGF) de $P(z)$ et $K(z)$ et donner les valeurs de $P_n = \mathrm{card}(\mathcal{P}_n)$ et $K_n = \mathrm{card}(\mathcal{K}_n)$.

Q2. Soit cycle(σ) le nombre de cycles dans la décomposition de σ. Partant de la description des permutations enrichie par la marque u qui repère les cycles,

$$\mathcal{P} = \textsc{Set}(u\,\textsc{Cyc}(\mathcal{Z})),$$

en déduire que la série génératrice bivariée du nombre $P_{n,k}$ de permutations ayant k cycles vérifie

$$P(z,u) \equiv \sum_{n,k} P_{n,k} u^k \frac{z^n}{n!} = \exp\left(u \log \frac{1}{1-z}\right).$$

Donner une expression plus compacte de $P(z,u)$.

Q3. Montrer que

$$\sum_k P_{n,k} u^k = u(u+1)(u+2)\cdots(u+n-1).$$

Déterminer l'espérance du nombre de cycles (cycle(σ)) sur l'ensemble des permutations de taille n. (Chaque permutation a probabilité $1/n!$.) Indiquer le principe du calcul de la variance et de l'écart type. On notera systématiquement H_n le nombre harmonique : $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$.

— *Permutation en place d'un tableau* —

On se donne un tableau $T[1\,..\,n]$ de n cases susceptible de contenir des données d'un certain type D ainsi qu'une permutation σ de taille n. On ne veut utiliser qu'un seul registre auxiliaire (ou variable) X qui peut contenir n'importe quel élément de D. Les opérations permises sont toutes les affectations entre cases du tableau et le registre :

$$T[i] := T[j]; \qquad T[i] := X; \qquad X := T[j];$$

pour des valeurs numériques quelconques de $i, j \in [1\,..\,n]$. L'objectif est, connaissant σ (qui est donné) de construire un programme Π_σ tel qu'à la fin de son exécution la valeur notée $T^\star[k]$ de la k-ième case du tableau vérifie pour tout k (avec $1 \le k \le n$)

$$T^\star[k] = T[\sigma(k)],$$

où $T[x]$ dans le membre droit est la valeur de la x-ième case avant exécution de Π_σ.

Q4. On considère $n = 2$ et $\sigma = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$. Donner la séquence bien connue de 3 affectations qui réalise l'échange des valeurs des cases 1 et 2 du tableau (au moyen du registre X).

Q5. On considère $n = 3$ et $\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$. Décrire le programme Π_σ dans ce cas. Combien d'affectations suffisent-elles pour effectuer cette permutation circulaire ? Généraliser au cas d'une permutation circulaire quelconque.

Q6. On considère $n = 4$ et $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}$. Combien d'affectations suffisent-elles pour réaliser Π_σ ? Même question pour $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$.

On note $\mathrm{fix}(\sigma)$ le nombre de points fixes de σ, c'est-à-dire le nombre d'indices j tels que $\sigma(j) = j$. Décrire une stratégie générale de construction de Π_σ. Soit $\mathrm{coût}(\sigma)$ le coût mesuré en nombre d'affectations de Π_σ. Exprimer ce coût en fonction des quantités suivantes : $n = |\sigma|, \quad \mathrm{cycle}(\sigma), \quad \mathrm{fix}(\sigma)$.

Q7. On marque désormais par v les points fixes d'une permutation. En s'aidant de la spécification des permutations marquée comme suit

$$\mathcal{Q} = \mathrm{S{\scriptsize ET}}\left(v\mathcal{Z} + \mathrm{C{\scriptsize YC}}_{\geq 2}(\mathcal{Z})\right),$$

déterminer une expression de

$$Q(z, v) \equiv \sum_{n,k} Q_{n,k} v^k \frac{z^n}{n!},$$

où $Q_{n,k}$ est le nombre de permutations de n ayant k points fixes.

Q8. Montrer à partir de ce qui précède que l'espérance du paramètre $\mathrm{coût}(\sigma)$ prise sur l'ensemble des permutations de taille n vaut

$$n + H_n - 1.$$

Donner les deux premiers termes asymptotiques de l'xespérance lorsque $n \to \infty$.

— *Détermination du maximum d'un tableau* —

On se donne désormais un tableau $U[1 \,..\, n]$ contenant une permutation des entiers $1, \ldots, n$. On considère l'algorithme suivant qui détermine la position (p) du maximum des éléments du tableau :

Algorithme `CalculDuMaximum`
- $M := -1\,;\, p := 0\,;$
- for j from 1 to n do
  - if $U[j] > M$ then $M := U[j]\,;\, p := j\,;$
- return(p).

On propose d'effectuer l'analyse en moyenne de l'algorithme. Pour cela on suppose que le tableau U est rempli par une permutation aléatoire de $[1 \,..\, n]$, chaque permutation étant choisie avec probabilité $1/n!$. La fonction de coût à analyser est le nombre de fois où la condition "$U[j] > M$" est vérifiée, c'est-à-dire le nombre de fois où la paire d'instructions "$M := U[j]\,;\, p := j\,;$" est exécutée.

Q9. Donner le cas le pire et le cas le meilleur du coût de l'algorithme, vis-à-vis de la fonction de coût étudiée.

Vérifier brièvement que, dans la décomposition (DECOMP) rappelée en début d'énoncé, le produit des r facteurs commute. Vérifier également qu'un τ qui vaut $[i_1, \ldots, i_\ell]$ peut se représenter de ℓ manières équivalentes.

Q10. On appelle *leader* d'une permutation circulaire τ le maximum des valeurs qui apparaissent dans l'ecriture $\tau = [i_1, \ldots, i_\ell]$ sous forme de liste circulaire. Dans la décomposition $\sigma = \tau_1 \circ \cdots \circ \tau_r$, on représente d'abord chaque τ_j en plaçant le leader en première position ; puis on ordonne entre eux dans le produit de composition tous les τ_j selon les valeurs croissantes de leur leader.

Lorsqu'on lit les éléments de la décomposition ainsi réorganisée en effaçant les symboles "$\circ$"et "$[,]$", on obtient une suite $w_1 w_2 \cdots w_n$ d'éléments distincts qui peut s'interpréter comme une nouvelle permutation $\sigma' = \begin{pmatrix} 1 & 2 & \cdots & n \\ w_1 & w_2 & \cdots & w_n \end{pmatrix}$. On désigne par $\lambda(\sigma)$ ce σ' associé à σ. Montrer que λ est une bijection de $\mathcal{P}_n$ sur lui-même.

Soit $R_{n,k}$ le nombre de permutations de taille n telles que l'algorithme donné ci-dessus, `CalculDuMaximum`, nécessite k affectations. Montrer l'égalité

$$R_{n,k} = Q_{n,k},$$

pour tous n, k. En déduire que le nombre moyen d'affectations effectuées par `CalculDuMaximum` vaut le nombre harmonique H_n.

Problème II
Bornes de codage

L'objectif de ce problème est d'effectuer le lien entre la combinatoire analytique et diverses bornes de codage qui interviennent dans deux types de problèmes : (*i*) le codage succint d'objets structurés par des mots binaires quelconques ; (*ii*) le codage de mots binaires quelconques sur un canal de communication contraint.

On considère deux ensembles finis A et C dont les cardinalités respectives sont notées $a = \mathrm{card}(A)$ et $c = \mathrm{card}(C)$. On dit que A est *codable* par C s'il existe une fonction $\phi : A \mapsto C$ telle que ϕ est injective (c'est-à-dire que $\phi(x) = \phi(y)$ implique que $x = y$; ou encore deux éléments distincts ont des images distinctes). On dit qu'une telle fonction ϕ est un *codage* et l'on fait référence à C comme l'*ensemble des codes*.

Q1. Justifier brièvement l'assertion suivante : Une condition nécessaire et suffisante sur les entiers a et c pour que A soit codable par C est que soit vérifiée l'inégalité fondamentale : $\boxed{a \leq c}$.

— Codage d'objets structurés —

Q2. On prend pour $\mathcal{A}$ l'ensemble $\mathcal{F}_n$ des objets de taille n dans une classe non-étiquetée $\mathcal{F}$. On prend pour ensemble des codes l'ensemble $C = \{0,1\}^\ell$.

Soit $F(z)$ la fonction génératrice ordinaire (ou OGF) $F(z) = \sum F_n z^n$ et soit R son rayon de convergence. On suppose désormais R fini et différent de 0 et l'on pose

$$\lambda := \log_2 \frac{1}{R}.$$

On choisit un réel $\epsilon > 0$ aussi petit que l'on veut. Montrer qu'un codage par $C = \{0,1\}^\ell$ est possible si

$$\ell \geq (\lambda + \epsilon)n,$$

pour tout n supérieur à un certain seuil n_0.

Q3. Montrer que, pour une infinité de valeurs de n, il n'existe pas de codage vérifiant

$$\ell \leq (\lambda - \epsilon)n.$$

Par la suite, le *taux de codage* est défini comme le rapport entre la taille de l'objet d'origine et la longueur du mot binaire qui le code.

Q4. On prend pour ensemble des codes l'ensemble $C = \{0,1\}^\ell$ des mots binaires de longueur ℓ. Soit $\mathcal{G}_n$ l'ensemble des arbres (enracinés, plans, non-étiquetés, tous les degrés de sommet étant permis) comprenant n sommets. On rappelle que $G_{n+1} = \mathrm{card}(\mathcal{G}_{n+1})$ vaut le nombre de Catalan $\frac{1}{n+1}\binom{2n}{n}$. Établir sans long calcul que, pour tout n assez grand, le codage de $A = \mathcal{G}_{n+1}$ par C est possible si

$$\ell \geq 2.01n.$$

Établir que, pour tout n assez grand, le codage de $A = \mathcal{G}_{n+1}$ par C est impossible si

$$\ell \leq 1.99n.$$

(On pourra par exemple utiliser la formule de Stirling : $n! \sim n^n e^{-n}\sqrt{2\pi n}$.) Comment ce résultat se compare-t-il à celui de la question **Q3** ?
Pouvez-vous décrire brièvement le principe d'un codage effectif qui atteigne asymptotiquement le taux de codage 2 ?

Q5. Une société de service doit stocker un grand nombre d'arbres (enracinés, plans, non-étiquetés, tous les degrés de sommet étant permis) qui sont toujours de hauteur ≤ 2. Soit $\mathcal{H}$ la famille de ces arbres. Justifier brièvement que $\mathcal{H}$ est spécifiée par

$$\mathcal{H} = \mathcal{Z} \times \textsc{Seq}(\mathcal{Z} \times \textsc{Seq}(\mathcal{Z})),$$

et calculer la fonction génératrice (OGF) correspondante. Déterminer le meilleur taux de codage qui peut être atteint dans ce cas. Pouvez-vous donner une description explicite d'un tel codage ?

Q6. Décrire à deux décimales la valeur du meilleur taux de codage qui peut être atteint pour des arbres de taille de hauteur ≤ 3 et de taille suffisamment grande. Noter la valeur numérique suivante : $\log_2 \frac{3+\sqrt{5}}{2} = 1.388\cdots$.
Discuter brièvement le cas d'arbres de hauteur $\leq h$: le meilleur taux de codage est-il un réel calculable ? (On ne demande pas un calcul explicite.)

— Codage par mots contraints —

Alice veut échanger des messages binaires de longueur n avec Bob sur un canal qui achemine des bits mais interrompt la communication dès qu'une suite de trois lettres $\mathsf{1}$ consécutives est détectée. Elle doit donc coder ses messages pris dans $A = \{0, 1\}^n$ par des mots de C qui respectent la contrainte du canal.

Q7. Alice et Bob prennent $C = \mathcal{E}_m$ où $\mathcal{E}$ est le langage des mots binaires ne comportant pas le motif $\mathsf{111}$ (consécutivement). Donner une description de $\mathcal{E}$ par les constructions de $(+, \times, \textsc{Seq})$ appliquées aux atomes $\mathsf{0}$ et $\mathsf{1}$ (ou, si l'on préfère, une expression régulière non ambiguë écrite avec $(+, \cdot, \star)$). Quelle est la fonction génératrice ordinaire $E(z)$ de $\mathcal{E}$?

On pourra s'aider du fait que l'ensemble de tous les mots peut être décrit par

$$\textsc{Seq}(\mathsf{1}) \times \textsc{Seq}(\mathsf{0}\,\textsc{Seq}(\mathsf{1})) \cong \mathsf{1}^\star \cdot (\mathsf{0}\mathsf{1}^\star)^\star.$$

Q8. Comment Alice et Bob doivent-ils choisir m pour pouvoir communiquer ? La communication est-elle possible si $m = 1.1n$? Est-elle possible si $m = 1.2n$? On indique que la seule racine réelle du polynôme $1 - z - z^2 - z^3$ est $\alpha = 0.54368\ldots$ et que $1/\log_2 \alpha = 1.137466\cdots$.

Q9. Un fournisseur d'accès facture 1€ (1 Euro) pour chaque bit transmis sur le canal qui relie Alice a Bob. Ceux-ci décident qu'ils ne veulent pas payer plus de 1.25€ par bit du message d'origine mais doivent convenir d'un algorithme effectif de codage. Bob propose alors de procéder comme suit :
– les messages sont décomposés par blocs de 2 bits ;
– ces blocs sont ensuite retranscrits selon la table

$$\mathsf{00} \mapsto \mathsf{000}, \quad \mathsf{01} \mapsto \mathsf{001}, \quad \mathsf{10} \mapsto \mathsf{010}, \quad \mathsf{11} \mapsto \mathsf{011}.$$

Ce schéma est-il correct vis à vis des contraintes du canal ? Satisfait-il aux conditions de coût voulues ?

Q10. Alice observe propose alors d'utiliser le langage :

$$\mathcal{L} = \mathsf{0} \cdot (\epsilon + \mathsf{1} + \mathsf{11}) \cdot (\mathsf{0}(\epsilon + \mathsf{1} + \mathsf{11}))^\star \cong \mathsf{0}(\epsilon + \mathsf{1} + \mathsf{11}) \times \textsc{Seq}\,(\mathsf{0}(\epsilon + \mathsf{1} + \mathsf{11}))$$

pour transcrire des blocs de $\{0, 1\}^r$ en mots distincts de $\mathcal{L}_s$. ($\mathcal{L}_s$ est l'ensemble des mots de longueur s dans $\mathcal{L}$.) Elle appelle un tel procédé un codage $A(r, s)$.

Donner une inégalité liant r et $L_s = \mathrm{card}(\mathcal{L}_s)$ pour qu'un tel procédé existe. Justifier qu'il vérifie alors la contrainte d'utilisation du canal.

Alice observe le développement suivant de l'OGF $L(z)$:

$$L(z) = z + 2z^2 + 4z^3 + \cdots + 149z^9 + 274z^{10} + 504z^{11} + \cdots .$$

Elle en déduit l'existence d'un code $A(r, s)$ satisfaisant de plus à la condition de coût. Quelles sont les valeurs de r et s obtenues par Alice ? Quelle est la taille de la table de transcription ?