TD d'algorithmique numéro 3 : Recherche de l'élément majoritaire
Olivier Bodini et Natacha Portier
lundi 8 octobre 2001

Question 1 ⋆ on atteint la complexité optimale pour la majorité ⋆

On change la manière de voir les choses. On a un ensemble de n balles et on cherche le cas échéant s'il y a une couleur majoritaire parmi les balles.

Question 1,1

Supposons que les balles soient rangées en file sur une étagère, de manière à n'avoir jamais deux balles de la même couleur à côté. Que peut-on en déduire sur le nombre maximal de balles de la même couleur ?

On a un ensemble de n balles, une étagère vide où on peut les ranger en file et une corbeille vide. Considérons l'algorithme suivant :

Phase 1 :

Prendre les balles une par une pour les ranger sur l'étagère ou dans la corbeille. SI la balle n'est pas de la même couleur que la dernière balle sur l'étagère, la ranger à côté, et si de plus la corbeille n'est pas vide, prendre une balle dans la corbeille et la ranger à côté sur l'étagère. SINON, c'est à dire si la balle est de la même couleur que la dernière balle sur l'étagère, la mettre dans la corbeille.

Phase 2 :

Soit C la couleur de la dernière balle sur l'étagère à la fin de la phase 1. On compare successivement la couleur de la dernière balle sur l'étagère avec C. SI la couleur est la même on jette les deux dernières balles sur l'étagère, sauf s'il n'en reste qu'une, auquel cas on la met dans la corbeille. SINON on la jette et on jette une des balles de la corbeille, sauf si la corbeille est déjà vide auquel cas on s'arrête en décrétant qu'il n'y a pas de couleur majoritaire. Quand on a épuisé toutes les balles sur l'étagère, on regarde le contenu de la corbeille. Si elle est vide alors il n'y a pas de couleur majoritaire, et si elle contient au moins une balle alors C est la couleur majoritaire.

Question 1,2 ⋆ correction de l'algorithme ⋆

Montrer qu'à tout moment de la phase 1, toutes les balles éventuelles de la corbeille ont la couleur de la dernière balle de l'étagère. En déduire que s'il y a une couleur dominante alors c'est C.

Prouver la correction de l'algorithme.

Question 1,3 ⋆ complexité ⋆

Donner la complexité dans le pire des cas en nombre de comparaisons de couleurs de balles.

Question 2 ⋆ Optimalité ⋆

On considère un algorithme de majorité pour les couleurs de n balles. Le but est de regarder faire l'algorithme, en choisissant au fur et à mesure les couleurs des balles pour que l'algorithme ait le maximum de travail (tout en restant cohérent dans le choix des couleurs). On obtiendra ainsi une borne inférieure de complexité pour un algorithme de majorité. C'est la technique de l'"adversaire".

À tout moment de l'algorithme, on aura une partition des balles en deux ensembles : l'arène et les gradins. L'arène contient un certain nombre de composantes connexes de deux sortes : les binômes et les troupeaux. Un binôme est un ensemble de deux balles pour lesquelles l'algorithme a déjà testé si elles étaient de la même couleur et a répondu non. Un troupeau est un ensemble non vide de balles de la même couleur, connectées par des tests de l'algorithme. Ainsi, un troupeau avec k éléments a subi au moins $k-1$ comparaisons de couleurs entre ses membres. Soient B le nombre de binômes et T le nombre de troupeaux. Soient g le nombre d'éléments dans les gradins et t le nombre total d'éléments dans tous les troupeaux. Enfin, soit $m = \lfloor n/2 \rfloor + 1$ le "nombre de majorité".

Au début de l'algorithme toutes les balles sont des troupeaux à un élément. La stratégie de l'adversaire est la suivante. L'algorithme effectue un test $couleur(x) = couleur(y)$?

1. Si x ou y sont dans les gradins, la réponse est non.
2. Si x (resp. y) est dans un binôme, la réponse est non et x (resp. y) est envoyé dans les gradins alors que l'élément restant devient un troupeau singleton.
3. Si x et y sont dans le même troupeau la réponse est oui.
4. Si x et y sont dans des troupeaux différents alors cela dépend de $d = B + t$.
   (a) $d > m$: cela signifie que les balles sont dans des troupeaux singletons. La réponse est non et les balles deviennent un nouveau binôme.
   (b) $d = m$: la réponse est oui et les troupeaux de x et y fusionnent.

Question 2,1

Vérifier que les 4 cas précédents traitent tous les cas possibles.

Montrer qu'à tout moment $d \geq m$ et que si $d > m$ alors tous les troupeaux sont des singletons.

Question 2,2

Montrer qu'à tout moment les deux coloriages suivants sont consistants avec les réponses de l'adversaire :

1. Toutes les balles sont de couleurs différentes sauf celles qui sont dans un même troupeau.
2. Une même couleur est attribuée à toutes les balles de tous les troupeaux et à une balle de chaque binôme. Les balles restantes ont chacune une couleur distincte.

Question 2,3

Montrer que si un algorithme correct s'arrête alors l'arène ne contient qu'une seule composante connexe qui est un troupeau de taille m.

Question 2,4

À tout moment le nombre de comparaisons inégales effectuées par l'algorithme est au moins $2g + B$ et le nombre de comparaisons égales est au moins $t - T$.

Question 2,5

Considérons un algorithme qui résout la majorité. Montrer qu'il existe une donnée pour laquelle l'algorithme effectue au moins $2(n - m) = 2\lceil n/2 \rceil - 1$ comparaisons d'inégalité et au moins $\lfloor n/2 \rfloor$ comparaisons d'égalité, et donc au moins au total $3\lceil n/2 \rceil - 2$ comparaisons.

Question 3,1 ⋆ matrices de Tœplitz ⋆

Une *matrice de Tœplitz* est une matrice $n \times n$ $(a_{i,j})$ telle que $a_{i,j} = a_{i-1,j-1}$ pour $2 \leq i, j \leq n$.

La somme de deux Tœplitz est-elle Tœplitz ? Et le produit ?

Question 3,2

Comment additionner deux matrices de Tœplitz en $O(n)$?

Question 3,3

Comment calculer le produit d'une matrice $n \times n$ de Tœplitz par un vecteur de longueur n ? Quelle est la complexité de l'algorithme ?

Question 4 ⋆ location de skis ⋆

Un groupe de n étudiants veut louer des skis dans un magasin de location. La boutique dispose de m paires (il y en a suffisemment). Idéalement chacun devrait avoir une paire de ski de même taille que lui. En pratique on cherche un algorithme pour minimiser la somme des valeurs absolues des différences (c'est la distance 1).

Question 4,1

Montrer que la fonction qui à la taille d'une personne associe la taille de ses skis peut être une fonction croissante.

En déduire le cas n=m et la complexité dans ce cas.

Question 4,2

On suppose les skieurs et les skis classés par taille croissante. Soit $A[i, j]$ le minimum de la fonction à minimiser si on considère les i premiers étudiants et les j premières paires de ski, avec $j \geq i + 1$.

Donner une relation de récurrence pour A, en déduire un algorithme et sa complexité en m et n.

Question 4,3

Montrer que lorsque $n^2 = o(m)$, et si on suppose les skis déjà classés par ordre croissant, on peut avoir une meilleure complexité en restreignant l'ensemble des skis à considérer.