ENSEIRB

Filière Informatique 1ère année 2003/2004

UV I1D :Méthodes Statistiques Module IS1 01 :Probabilités et Statistiques

Durée :3 h le 11 /06/2004

Documents de cours et TD autorisés.

Avertissement: Il sera tenu le plus grand compte de la justification des réponses et de la présentation .

Tous les algorithmes demandés seront (naturellement) écrits en L.D.A , clairs et bien commentés .

E est un ensemble d'éléments avec une relation d'ordre total, notée classiquement $\leq$. L'algorithme Rand_Tri reçoit en entrée une suite de N éléments de E et retourne en sortie la suite triée suivant l'ordre croissant .

Principe de l'algorithme Rand_Tri : un élément de la suite (appelé : pivot) est choisit au ''hasard'' (c'est à dire que chaque élément de la suite a la même probabilité d'être choisi) ; on compare le pivot avec tous les éléments de la suite , ceux qui sont $\leq$ au pivot sont placés avant le pivot et les autres après le pivot ; on obtient ainsi 2 sous suites (éventuellement l'une des deux peut être vide) . On itère ce procédé sur chaque sous suite ainsi formée, tant qu'elles comportent plus d'un élément . Lorsque l'algorithme s'arrête la suite de départ est triée suivant l'ordre croissant .
On s'intéresse à la variable aléatoire N_C nombre de comparaisons effectuées au cours de l'exécution de l'algorithme .

Question préliminaire : le nombre de comparaisons effectuées au cours de l'exécution de l'algorithme dépend naturellement de l'ordre initial de la suite d'entrée, qui n'est pas connu, notons L la loi de l'ordre de la suite d'entrée . Montrer qu'en fait la loi de N_C ne dépend pas de la loi L ; c'est à dire précisément, que : pour tout entier k
Prob(N_C = k , connaissant l'ordre d'entrée) = Prob(N_C = k)

I Calcul de l'espérance de N_C ,notée E(N_C)

1) on note x_i i=1…N les éléments de la suite en sortie de l'algorithme (c.à.d.. triée suivant l'ordre croissant) ; soient i et j tels que $1 \leq i < j \leq N$;

- Dans le cas où $j>i+1$, il existe des éléments x_k entre x_i et x_j ; montrer que la Probabilité qu'au cours de l'exécution de l'algorithme x_i ou x_j aient été choisis comme pivot avant l'un quelconque des x_k pour $i<k<j$, vaut $2/(j-i+1)$. En déduire la Probabilité qu'au cours de l'exécution de l'algorithme, il y ait eu une comparaison entre x_i et x_j .
- Dans le cas où $j=i+1$, la méthode de calcul ci-dessus n'a plus de sens, cependant, vérifier directement que la formule que vous avez trouvée pour la Probabilité est encore valable .

SUITE AU VERSO

2) on note $X_{i,j}$ la variable aléatoire caractéristique de l'événement : ''au cours de l'exécution de l'algorithme, il y a eu une comparaison entre x_i et x_j '' .Calculer l'espérance de $X_{i,j}$ et sa variance . Trouver une relation **très simple** entre N_C et les $X_{i,j}$ .A l'aide de cette relation trouver une formule qui donne E(N_C) en fonction de N (vous ne chercherez pas à simplifier cette formule qui contient une double sommation)

3) écrire un algorithme qui calcule E(N_C) .

II Estimation de la variance de N_C , notée Var(N_C)

Maintenant , on suppose que N le nombre d'éléments des suites triées , est un '' grand nombre '' du point de vue des statistiques .

1) les variables aléatoires $X_{i,j}$ sont elles deux à deux indépendantes ?
Pour calculer Var(N_C) , pourquoi ne peut-on utiliser la relation entre N_C et les $X_{i,j}$ d'une manière analogue à ce qui a été fait dans I 2) ?

2) on suppose disponible une fonction appelée rand_tri qui lance l'exécution de l'algorithme Rand_Tri et retourne le nombre de comparaisons effectuées au cours de cette exécution ; écrire un algorithme (après avoir justifié son principe) qui donne une estimation de Var(N_C) .

3) vous supposerez que le paramètre taille_échantillon (que vous avez du introduire dans l'algorithme écrit en 2)) est supérieur à 100 , compléter votre algorithme pour qu'il donne aussi un intervalle de confiance de la forme classique (0 , b) pour Var(N_C) , au seuil de 0.05 .

4) on considère la condition : Probabilité de
(valeur absolue de [ESTIMATEUR de Var(N_C) - Var(N_C)] / Var(N_C) < 0.01)
> 0.95
écrire un algorithme (en le justifiant) qui détermine la valeur minimum du paramètre taille_échantillon qui permet d'obtenir la condition ci dessus .
Ecrire une phrase simple qui traduit intuitivement la condition considérée .