G. Hunault
Licence Informatique

Angers, octobre 2002

Controle continu pour l'Option de Probabilités et Statistiques

Modalités du Controle

Le controle continu comprend deux petits projets à rendre, l'un concernant la partie Probabilités, l'autre la partie Statistiques. Chaque étudiant(e) doit rendre les deux projets. Chaque projet comporte une partie programmation et une partie rédaction **qui ne comptent pas à part égale** dans la note.

La rédaction se fait au choix mais exclusivement sous *Word* ou sous *Latex*. La programmation est effectuée au choix mais exclusivement en *Awk*, *Apl*, *Rexx*, *C*, *C++*, *Perl*, *Tcl*, *Java* ou *Pascal*. On pourra ne rendre qu'un seul texte qui contiendra les explications sur les programmes des deux projets. Tous les documents (explications, algorithmes et programmes) seront uniquement transmis par *mail* sous forme de fichier informatique. Ce fichier, unique, sera au format .zip ou .tar et les fichiers y seront archivés sans sous-répertoire. Ce fichier devra être remis au plus tard le

mercredi 11 décembre, à 11 h dernier délai

Pour éviter toute ambiguité, tous les noms de fichiers commenceront par les initiales de l'élève. Il sera possible de vérifier auprès de l'enseignant le lundi 16 décembre entre 14 et 18 h que l'archive a bien été reçue et que les fichiers ont bien été décompressés et qu'ils sont lisibles, compilables etc.

1. Projet de Probabilités

La société "La Française des jeux" organise et gère le loto national. Nous rappelons qu'un tirage correspond à 6 nombres entiers distincts compris entre 1 et 49. Vous devez écrire un algorithme qui simule n tirages, n étant passé en paramètre, n pouvant varier de 1 à 99999. L'algorithme devra écrire les tirages successifs dans un fichier nommé tirages.lst puis mettre dans le fichier loto.res les numéros rangés par ordre décroissant d'apparition. Par exemple si on effectue $n = 2$ tirages et qu'on obtient les tirages

```
6  7 11 17 26 35
8 10 11 25 36 43
```

alors on aura comme résultat

```
NbTir   Frq (%)    NbEnt   Liste pour 2 tirages
 2      100         1       11
 1       50        10         6  7  8 10 17 25 26 35 36 43
 0        0        38         1  2  3  4  5  9 ...
```

Il vous faudra ensuite traduire l'algorithme dans un langage de votre choix parmi les langages cités puis rédiger la documentation de votre algorithme et de votre programme. Enfin, pour conclure ce projet de probabilités, vous essaierez de répondre aux questions suivantes en justifiant vos sources (ce pourra être des URL) :

- combien y a-t-il eu de tirages depuis le début du Loto national en France ?
- la simulation donne-t-elle "sensiblement" la même fréquence à chaque numéro ? que faut-il en déduire ?
- les résultats des fréquences par numéro donnés officiellement sont-ils "sensiblement" équivalents à vos résultats ? pourquoi ? comment le prouver ?
- cela vous permettra-t-il de gagner des millions d'euros aux prochains tirages ?

2. Projet de Statistiques

Les statistiques traitent des données. Une utilisation intelligente des données requiert parfois de choisir un ordre de traitement des données. C'est ce que vous devrez faire ici avec un affichage par somme de distances minimales entre plus proches voisins.

Soient c_1 et c_2 deux chaines de caractères de même longueur. On appelle indicateur de distance de *Hamming* entre les chaines noté $dh(c_1, c_2)$ le rapport du nombre de positions où les caractères sont différents au nombre de caractères en tout. Par exemple pour les deux chaines

```
ALBATROS
ALBATURD
```

la longueur commune est 8, les 3 dernières positions comportent des lettres différentes, donc le dh vaut 3/8 soit 0.375.

On appelle matrice de *Hamming* d'une liste de chaines c_k de longueur non nulle la matrice triangulaire inférieure $M_{i,j}$ tel que $M_{i,j} = dh(c_i, c_j)$.

On appelle ordre par somme de distances maximales entre plus proches voisins au sens de Hamming la liste des numéros de chaines présentée comme suit : la chaine qui suit la chaine numéro i est la chaine plus proche de i parmi les chaines restantes ; si plusieurs chaines réalisent le minimum, on regarde celle dont le[s] suivant[s] donnerai[en]t le minimum de distance.

Voici un exemple qui illustre cet ordre : soient 4 chaines initialement numérotées 1, 2, 3 et 4 dont la matrice de *Hamming* est

```
        1       2      3       4
----+-------------------------
1   + 0
2   + 0.4     0
3   + 0.2     0.5    0
4   + 0.3     0.4    0.6     0
```

L'indicateur de distance minimal est 0.2, réalisé par le couple (1,3). L'élément le plus proche de 1 est ensuite 4 (dh = 0.3) alors que que l'élément le plus proche de 3 est 2 (dh=0.5). L'ordre des chaines doit donc être

```
3   1   4   2
```

Pour réaliser le projet de statistiques vous viendrez dans un premier temps écrire un algorithme qui lit un fichier-texte dont le nom est passé en paramètre, sachant que ce fichier contient un nombre variable de chaines de même longueur à raison d'une chaine par ligne. L'algorithme doit ensuite calculer la matrice de *Hamming* de la liste des chaines et présenter les indicateurs de distance par ordre croissant de distance suivant le format D N L détaillé ci-dessous puis afficher l'ordre des chaines par somme de distances minimales entre plus proches voisins.

Vous n'oublierez pas qu'il peut y avoir plus de 10 chaines et que les numéros de chaines devront être bien cadrés, unité sous unité, dizaine sous dizaine... Vous pourrez arrondir les indicateurs de distance à 10^{-4} pour les affichages...

Le format D N L correspond à l'affichage des indicateurs de distance, au nombre de couples le réalisant puis à la liste des couples (par ordre croissant) qui le réalisent. Pour l'exemple précédent, ce sera donc

```
Ind.    Nb.      Couples
  0.2     1        1 - 3
  0.3     1        1 - 4
  0.4     2        1 - 2 ; 2 - 4
  0.5     1        2 - 3
  0.6     1        3 - 4
```

Vous viendrez ensuite traduire l'algorithme dans un langage de votre choix parmi les langages cités puis rédiger la documentation de votre algorithme et de votre programme. Vous rédigerez ensuite une page entière sur le ou les inventeurs de ce type de présentation, les intérêts (cachés) de cet affichage, la disponibilité de cet affichage dans les logiciels statistiques usuels sans oublier de citer précisément vos sources. Celles-ci pourront être des URL. Vous pourrez également discuter du choix du terme prudent "indicateur de distance" plutot que du terme distance et vous essaierez de trouver 4 chaines de longueur minimale dont la matrice de *Hamming* est celle de l'exemple illustratif.

Vous fournirez également l'affichage obtenu si on exécute votre programme en lui passant comme paramètres la chaine *strings.ham* qui est l'identificateur d'un fichier texte qui contient 14 chaines de longueur 10. Ce fichier est disponible à l'adresse

http://www.info.univ-angers.fr/pub/gh/wstat/strings.ham

ainsi que sur le disque K: des machines de la Faculté des Sciences, dans le sous-répertoire Stat_Ad.

Remarques concernant la rédaction

La note sera principalement fonction de la qualité de l'analyse du travail effectué, de l'exposé de la méthode et de la clarté de la rédaction. La valeur attribuée au programme ne représentera que 25 % de la note. Aucune interface n'est demandée, les programmes devront être exécutables en ligne de commande. En cas de langage compilé, on fournira impérativement le programme-source et éventuellement l'exécutable produit. On précisera, si besoin est, la commande de compilation. En cas de langage interprété, on fournira impérativement le programme-source. Dans tous les cas de figures, on précisera la version du langage choisi, le type d'ordinateur et le système d'exploitation utilisés. Aucun langage autre que cité ne devra être utilisé, aucun traitement de textes autre que cité ne devra être utilisé. Tout fichier qui ne correspondra pas aux initiales de l'élève ne sera pas lu. En particulier, les fichiers nommés uniquement proba, algo, stat,projet, doc ne seront pas lus.

Un programme même incomplet mais commenté et bien indenté sera au moins autant apprécié qu'un programme qui "marche" mais sans commentaire et illisible quant aux structures de flot (boucles, tests etc.).

Les identificateurs auront tous au moins 4 caractères. Les algorithmes utiliseront la syntaxe de *Galg*. On fournira d'ailleurs non pas le texte original des algorithmes mais leurs résultats d'analyse par *Galg* avec le numéro de ligne, le numéro d'instruction, l'indication de profondeur etc. et le fichier des tableaux, variables et modules.

3. Références

Ce texte, de même que le cours de *Probabilités et Statistiques* pour la licence informatique, les T.D. correspondants (avec solution) et la procédure pour envoyer le mini-projet peut être récupéré à partir de l'adresse

`http://www.info.univ-angers.fr/pub/gh/wstat/`

Un manuel élémentaire d'*Algorithmiques raisonnées* et les références à *Galg* sont disponibles depuis la page

`http://www.info.univ-angers.fr/pub/gh/wals/`

De plus, il est possible de tester si le projet a été correctement reçu grâce à la page

`http://www.info.univ-angers.fr/pub/gh/wstat/zipdirpli.php3`