CONCOURS D'ADMISSION DE 1992

MATHEMATIQUES II

Option générale

Jeudi 7 mai 1992 de 8h à 12h

Dans tout le problème, on désigne par n un nombre entier naturel non nul.

PARTIE I

On se propose dans cette partie de calculer pour tout entier $p \geq 1$ la somme:

$$S_n(p) = \sum_{k=1}^{p} k^n = 1^n + 2^n + \cdots + p^n.$$

Soient $\mathbb{R}[X]$ l'espace vectoriel des fonctions polynômes à coefficients réels et (P_k) la suite de fonctions polynômes définies par $P_0(x) = 1$ et, si $k \geq 1$, par:

$$P_k(x) = \frac{x(x-1) \ \ldots \ (x-k+1)}{k!} = \frac{1}{k!} \prod_{j=0}^{k-1} (x-j).$$

On considère l'application Δ de $\mathbb{R}[X]$ dans lui-même associant à la fonction polynôme P la fonction polynôme ΔP définie par: $(\Delta P)(x) = P(x+1) - P(x)$.

a) Prouver que Δ est un endomorphisme de $\mathbb{R}[X]$, puis déterminer son noyau.
b) Soient j, k des entiers naturels. On rappelle que $\Delta^j = \Delta \circ \Delta \circ \ldots \circ \Delta$ (j fois), et Δ^0 désigne par convention l'application identité de $\mathbb{R}[X]$.
Déterminer ΔP_k, puis $\Delta^j P_k$ en distinguant les cas $j \leq k$ et $j > k$.
c) Prouver que $(P_0, P_1, \ldots, P_n)$ est une base du sous-espace de $\mathbb{R}[X]$ constitué des fonctions polynômes de degré inférieur ou égal à n, et établir la formule suivante pour toute fonction polynôme P de degré inférieur ou égal à n:

$$P(x) = \sum_{k=0}^{n} (\Delta^k P)(0) . P_k(x).$$

d) En déduire que Δ est surjectif, puis, pour toute fonction polynôme P de degré n, établir l'existence d'une unique fonction polynôme Q, de degré n+1, telle que:

$$Q(x+1) - Q(x) = P(x) \quad et \quad Q(0) = 0.$$

On note Q_n la fonction polynôme telle que $Q_n(x+1) - Q_n(x) = x^n$ et $Q_n(0) = 0$.
e) Exprimer $S_n(p)$ à l'aide de Q_n et de p, et expliciter Q_n et $S_n(p)$ pour $n \leq 3$.

PARTIE II

On établit dans cette partie quelques résultats probabilistes.

1°) Fonction génératrice d'une variable aléatoire.

On considère un entier naturel p et une variable aléatoire X à valeurs dans l'ensemble {0, 1, ... , p}.

On lui associe la fonction suivante, dite fonction génératrice de X:

$$t \to G_X(t) = \sum_{k=0}^{p} P(X=k)t^k.$$

Déterminer la fonction génératrice de X lorsque:

a) X suit une loi de Bernoulli de paramètre λ (λ réel tel que $0 < \lambda < 1$).

b) X suit une loi binômiale de paramètres n et λ (λ réel tel que $0 < \lambda < 1$).

2°) Fonction génératrice d'une somme de variables aléatoires indépendantes.

On considère désormais des entiers naturels $p_1, p_2, \dots, p_n$ et des variables aléatoires $X_1, X_2, \dots, X_n$ définies sur un même espace probabilisé, indépendantes et à valeurs dans $\{0, 1, \dots, p_1\}$, $\{0, 1, \dots, p_2\}$, ... , $\{0, 1, \dots, p_n\}$ respectivement. On note alors: $Y_n = X_1 + X_2 + \dots + X_n$.

a) Etablir que la fonction génératrice de $Y_2 = X_1 + X_2$ est le produit des fonctions génératrices de X_1 et de X_2.

b) Exprimer la fonction génératrice de $Y_n = X_1 + X_2 + \dots + X_n$ à l'aide des fonctions génératrices de X_1, de X_2, ... , de X_n.

Retrouver le résultat obtenu en 1°(b) à partir du résultat obtenu en 1°(a).

3°) Etude d'un cas particulier.

On suppose de plus que, pour tout entier k tel que $1 \le k \le n$, la variable aléatoire X_k suit une loi uniforme sur {0, 1, ... , k-1}, autrement dit:

$$P(X_k=0) = P(X_k=1) = \dots = P(X_k=k-1) = \frac{1}{k}.$$

a) Calculer l'espérance et la variance des variables aléatoires X_k ($1 \le k \le n$), puis de la somme $Y_n = X_1 + X_2 + \dots + X_n$.

On convient de noter la fonction génératrice G_n de Y_n sous la forme suivante:

$$G_n(t) = \frac{1}{n!} \sum_{k=0}^{\frac{n(n-1)}{2}} c_{n,k}\, t^k.$$

b) En exprimant G_{n+1} en fonction de G_n, déterminer $c_{n+1,k}$ en fonction des coefficients $c_{n,k}$, $c_{n,k-1}$, ... , $c_{n,k-n}$.

Par convention, on posera $c_{n,k} = 0$ lorsque $k < 0$ ou lorsque $k > n(n-1)/2$.

c) Montrer que les coefficients $c_{n,k}$ sont entiers naturels et calculer leur somme pour $0 \le k \le n(n-1)/2$.

PARTIE III

On considère un tableau s = (s[1], s[2], ... , s[n]) dans lequel toutes les variables sont de type entier et où les n entiers 1, 2, ... , n sont affectés dans un ordre quelconque aux n variables s[1], s[2], ... , s[n].
L'objet de cette partie est l'étude de la complexité de l'algorithme de tri suivant, visant à ranger dans l'ordre croissant les éléments de ce tableau s:

```
begin
for i:=1 to n-1 do
   for j:=1 to n-i do
     if s[j] > s[j+1] then  Echanger(s[j],  s[j+1]);
end;
```

(la procédure Echanger(x, y) permute les valeurs des variables x, y).

1°) Exemples de mise en œuvre de l'algorithme.
On suppose dans cette question que n = 3. Indiquer les valeurs contenues dans les variables s[1], s[2], s[3] après chaque appel de la procédure Echanger lorsque l'on affecte initialement à (s[1], s[2], s[3]) les six valeurs suivantes:
(1, 2, 3) ; (2, 3, 1) ; (3, 1, 2) ; (1, 3, 2) ; (3, 2, 1) ; (2, 1, 3).
On précisera à chaque fois le nombre total d'appels de la procédure Echanger.
(on présentera les résultats obtenus dans six tableaux distincts).

2°) Action de l'algorithme.
a) Déterminer la valeur contenue dans s[n] à l'issue du passage dans la boucle suivante:

```
for j:=1 to n-1 do
if s[j] > s[j+1] then  Echanger(s[j],  s[j+1]);
```

b) En déduire la valeur contenue en fin d'algorithme dans s[k] pour $1 \leq k \leq n$.

3°) Complexité de l'algorithme.
a) Exprimer en fonction de n le nombre de comparaisons d'entiers (>) effectuées au cours de l'algorithme.
b) Exprimer en fonction de n les nombres maximal et minimal d'appels de la procédure Echanger au cours de l'algorithme. On explicitera des tableaux s pour lesquels ces maximum et minimum sont effectivement atteints.

Dans la suite, on associe au tableau s et à tout entier k tel que $1 \leq k \leq n$:
- le nombre $U_k(s)$ des entiers i tels que $1 \leq i < k$ et $s[i] > s[k]$ (avec $U_1(s) = 0$).
- la somme $V_n(s) = U_1(s) + U_2(s) + \ldots + U_n(s)$.

On dit que $V_n(s)$ est le nombre des inversions du tableau s, autrement dit le nombre des couples d'entiers (i, k) tels que:

$$1 \leq i < k \leq n \text{ et } s[i] > s[k].$$

4°) Nombre d'appels de la procédure Echanger.

a) Lorsque s[j] > s[j+1], indiquer l'effet de l'échange des valeurs de s[j] et s[j+1] sur le nombre des inversions du tableau s = (s[1], s[2], ... , s[n]).
En déduire le nombre d'appels de la procédure Echanger lors du tri de s.

b) On suppose dans cette question que n = 3. Préciser $(U_1(s), U_2(s), U_3(s))$ et $V_3(s)$ lorsque l'on affecte initialement à s = (s[1], s[2], s[3]) les six valeurs:
(1, 2, 3) ; (2, 3, 1) ; (3, 1, 2) ; (1, 3, 2) ; (3, 2, 1) ; (2, 1, 3).

c) Rédiger un algorithme de calcul de $U_k(s)$ (où $1 \leq k \leq n$) et de $V_n(s)$.
(Les candidats sont tenus de rédiger les algorithmes en langage PASCAL, en conservant toutefois les noms des variables introduites dans l'énoncé).

On suppose dans la suite que les différentes façons d'affecter les n entiers 1, 2, ... , n aux n variables s[1], s[2], ... , s[n] sont équiprobables, et l'on note:
- S_n l'ensemble des différents tableaux s ainsi obtenus.
- Ω_n l'ensemble des n-uplets $(u_1, u_2, \ldots, u_n)$ d'entiers tels que $0 \leq u_k < k$ pour tout entier k tel que $1 \leq k \leq n$.

On considère $U_1, U_2, \ldots, U_n$ comme des variables aléatoires sur l'ensemble S_n.

5°) Bijectivité de l'application $s \rightarrow (U_1(s), U_2(s), \ldots, U_n(s))$.

a) Vérifier que l'application associant à tout tableau s de S_n le n-uplet $(U_1(s), U_2(s), \ldots, U_n(s))$ prend ses valeurs dans Ω_n.

b) Soit s un tableau de S_n tel que $U_1(s) = u_1, U_2(s) = u_2, \ldots, U_n(s) = u_n$.
- Déterminer s[n] en fonction de u_n.
- Expliquer de même comment l'on peut obtenir les valeurs de s[n-1], ... , s[1] en fonction de $u_{n-1}, \ldots, u_1$.

c) Montrer que l'application $s \rightarrow U(s)$ est une bijection de S_n sur Ω_n.
En déduire le nombre des tableaux s de S_n tels que $U_k(s) = u_k$ pour tout entier k tel que $1 \leq k \leq n$ et tout entier u_k tel que $0 \leq u_k < k$.

6°) Lois des variables aléatoires $U_1, U_2, \ldots, U_n$.

a) Déterminer la probabilité $P(U_1 = u_1, U_2 = u_2, \ldots, U_n = u_n)$ pour tout n-uplet $(u_1, u_2, \ldots, u_n)$ appartenant à Ω_n.

b) Déterminer la probabilité $P(U_k = u_k)$ pour tout entier k tel que $1 \leq k \leq n$ et tout entier u_k tel que $0 \leq u_k < k$.
En déduire la loi de U_k et l'indépendance des n variables aléatoires $U_1, \ldots, U_n$.

c) En déduire en fonction de n l'espérance et la variance du nombre d'appels de la procédure Echanger au cours de l'algorithme, et donner des équivalents simples de ces expressions lorsque n tend vers l'infini.

***