PARTIE I

Hors d'œuvre N°1

a) Soient $(P,Q) \in \mathbb{R}[X]$ et $\lambda \in \mathbb{R}$.

$\Delta(\lambda P+Q) = (\lambda P+Q)(X+1) - (\lambda P+Q)(X) = \lambda(P(X+1)-P(X)) + (Q(X+1)-Q(X)) = \lambda \Delta P + \Delta Q$

Ceci montre le caractère linéaire de Δ qui est clairement une application de $\mathbb{R}[X]$ dans lui-même.

Par conséquent Δ est un endomorphisme de $\mathbb{R}[X]$. Déterminons son noyau

Soit $P \in \mathrm{Ker}\,\Delta$. $P(X+1)-P(X)=0$; $P(X+1)=P(X)$

Supposons P non constant ; P admet au moins un zéro a dans $\mathbb{C}$.

$P(a+1)=P(a)=0$; une récurrence simple prouve alors que : $\forall n \in \mathbb{N},\ P(a+n)=0$; P admet alors une infinité de zéros ce qui n'est pas raisonnable pour un polynôme non constant donc non nul.

Donc si $P \in \mathrm{Ker}\,\Delta$, P est constant. Réciproquement si P est constant : $\Delta P = 0_{\mathbb{R}[X]}$ et $P \in \mathrm{Ker}\,\Delta$.

Finalement : $\underline{\mathrm{Ker}\,\Delta = \mathbb{R}_0[X]}$

Remarque .. On peut aussi obtenir ce résultat en remarquant que si P n'est pas constant : $\deg \Delta P = \deg P - 1$. Ce dernier résultat permet encore de montrer que :

$\forall n \in \mathbb{N}^*,\ \Delta(\mathbb{R}_n[X]) = \mathbb{R}_{n-1}[X]$. De ceci découle : $\Delta(\mathbb{R}[X]) = \mathbb{R}[X]$ et donc la surjectivité de Δ ... mais patience !

b) Soit $k \in \mathbb{N}^*\setminus\{1\}$. $\Delta P_k = P_k(X+1) - P_k(X) = \frac{1}{k!}\prod_{j=0}^{k-1}(X+1-j) - \frac{1}{k!}\prod_{j=0}^{k-1}(X-j)$

$$\Delta P_k = \frac{1}{k!}\left[\prod_{j=0}^{k-1}(X-j+1) - \prod_{j=1}^{k}(X-j+1)\right] = \frac{1}{k!}\left[\prod_{j=1}^{k-1}(X-j+1)\right]\left[(X+1)-(X-k+1)\right] = \frac{1}{k!} \times k \times \prod_{j=1}^{k-1}(X-j+1)$$

$$\Delta P_k = \frac{1}{(k-1)!}\prod_{j=1}^{k-1}(X-j+1) = \frac{1}{(k-1)!}\prod_{j=0}^{k-2}(X-j) = P_{k-1} \quad ; \quad \Delta P_k = P_{k-1}$$

Notons que ce résultat vaut encore pour $k=1$ car $\Delta P_1 = \Delta X = (X+1)-X = 1 = P_0$.

Finalement : $\underline{\forall k \in \mathbb{N}^*,\ \Delta P_k = P_{k-1}}$.

Soit $k \in \mathbb{N}$.

Une récurrence très simple(*) donne alors : $\forall j \in [\![0,k]\!],\ \Delta^j P_k = P_{k-j}$

(*) - OK pour $j=0$, $\Delta^0 P_k = P_k = P_{k-j}$
- OK pour $j \in [\![0,k-1]\!]$ donc $\Delta^{j+1}P_k = \Delta(\Delta^j P_k) = \Delta(P_{k-j}) = P_{k-j-1} = P_{k-(j+1)}$

En particulier $\Delta^k P_k = P_0 = 1$ et $\Delta^{k+1}P_k = \Delta(\Delta^k P_k) = \Delta 1 = 0_{\mathbb{R}[X]}$.

Donc si $j \in [\![k+1,+\infty[\![$: $\Delta^j P_k = \Delta^{j-k-1}(\Delta^{k+1}P_k) = \Delta^{j-k-1}(0_{\mathbb{R}[X]}) = 0_{\mathbb{R}[X]}$

En conclusion, si $k \in \mathbb{N}$,
$$\Delta^j P_k = \begin{cases} P_{k-j} & \text{si } j \in [\![0,k]\!] \\ 0_{\mathbb{R}[X]} & \text{si } j \in [\![k+1,+\infty[\![\end{cases}$$

c) $\forall k \in [\![0,n]\!]$, $\deg P_k = k$. $(P_0, P_1, \dots, P_n)$ est donc une famille libre de $\mathbb{R}_n[X]$ (polynômes de degrés échelonnés), cette famille étant de longueur $n+1 = \dim \mathbb{R}_n[X]$, c'est une base de $\mathbb{R}_n[X]$

Soit $P \in \mathbb{R}_n[X]$. $\exists (\lambda_0, \lambda_1, \dots, \lambda_n) \in \mathbb{R}^{n+1}$, $P = \sum_{k=0}^{n} \lambda_k P_k$.

$\forall j \in [\![0,n]\!]$, $\Delta^j P = \sum_{k=0}^{n} \lambda_k \Delta^j P_k = \sum_{k=j}^{n} \lambda_k P_{k-j}$. Notons encore que $P_\ell(0) = \begin{cases} 1 \text{ si } \ell = 0 \\ 0 \text{ si } \ell \in \mathbb{N}^* \end{cases}$

Donc $\forall j \in [\![0,n]\!]$, $(\Delta^j P)(0) = \sum_{k=j}^{n} \lambda_k P_{k-j}(0) = \lambda_j$. $\forall j \in [\![0,n]\!]$, $\lambda_j = (\Delta^j P)(0)$.

En conséquence.. $\forall P \in \mathbb{R}_n[X]$, $P = \sum_{k=0}^{n} (\Delta^k P)(0)\, P_k$

d) Soit $P \in \mathbb{R}[X]$. Montrons que : $\exists \varphi \in \mathbb{R}[X]$, $\Delta \varphi = P$.

$\exists n \in \mathbb{N}$, $P \in \mathbb{R}_n[X]$!

Donc $P = \sum_{k=0}^{n} \Delta^k P(0)\, P_k = \sum_{k=0}^{n} [(\Delta^k P)(0)]\, \Delta P_{k+1} = \Delta\left(\sum_{k=0}^{n} [(\Delta^k P)(0)]\, P_{k+1}\right)$

Posons $\varphi \overset{(1)}{=} \sum_{k=0}^{n} [(\Delta^k P)(0)]\, P_{k+1}$; $\varphi \in \mathbb{R}[X]$ et $\Delta \varphi = P$

Par conséquent : $\forall P \in \mathbb{R}[X]$, $\exists \varphi \in \mathbb{R}[X]$, $\Delta \varphi = P$; Δ est surjectif.

Notons que l'égalité (1) donne $\varphi(0) = \sum_{k=0}^{n} [(\Delta^k P)(0)]\, \underbrace{P_{k+1}(0)}_{=0} = 0$

Nous avons donc prouvé que : $\forall P \in \mathbb{R}[X]$, $\exists \varphi \in \mathbb{R}[X]$, $\Delta \varphi = P$ et $\varphi(0) = 0$.

Montrons l'unicité de ce φ

Soit $P \in \mathbb{R}[X]$. Supposons que $(\varphi, \hat{\varphi}) \in \mathbb{R}[X]^2$, $\Delta \varphi = P = \Delta \hat{\varphi}$ et $\varphi(0) = \hat{\varphi}(0) = 0$.

Alors $\Delta(\varphi - \hat{\varphi}) = 0_{\mathbb{R}[X]}$. $\varphi - \hat{\varphi} \in \operatorname{Ker} \Delta = \mathbb{R}_0[X]$. $\exists \lambda \in \mathbb{R}$, $\varphi - \hat{\varphi} = \lambda$.

$\lambda = (\varphi - \hat{\varphi})(0) = 0$; $\varphi = \hat{\varphi}$!

Finalement : $\forall P \in \mathbb{R}[X]$, $\exists! \varphi \in \mathbb{R}[X]$, $\Delta \varphi = P$ et $\varphi(0) = 0$

ou $\forall P \in \mathbb{R}[X]$, $\exists! \varphi \in \mathbb{R}[X]$, $\varphi(X+1) - \varphi(X) = P$ et $\varphi(0) = 0$.

e) $n \in \mathbb{N}^*$. $\varphi_n(X+1) - \varphi_n(X) = X^n$ et $\varphi_n(0) = 0$.

$\forall p \in \mathbb{N}^*$, $S_n(p) = \sum_{k=1}^{p} k^n = \sum_{k=1}^{p} (\varphi_n(k+1) - \varphi_n(k)) = \varphi_n(p+1) - \varphi_n(1) = \varphi_n(p+1) - (\varphi_n(0) + 0^n) = \varphi_n(p+1)$

$\forall p \in \mathbb{N}^*$, $S_n(p) = \varphi_n(p+1)$.

Remarque.. On est loin de la promesse initiale ("on se propose de calculer pour tout entier $p \geq 1$...")

d'après ce qui précède $\varphi_1 = \sum_{k=0}^{1} [(\Delta^k X)](0) P_{k+1}$; $\varphi_2 = \sum_{k=0}^{2} [(\Delta^k X^2)](0) P_{k+1}$ et $\varphi_3 = \sum_{k=0}^{3} (\Delta^k X^3)(0) P_{k+1}$

$(\Delta^0 X)(0) = \underline{0}$; $(\Delta^1 X)(0) = \underline{1}$; $\underline{\underline{\varphi_1 = P_2}}$.

$(\Delta^0 X^2)(0) = \underline{0}$; $(\Delta^1 X^2)(0) = \underline{1}$ $(\Delta X^2 = (X+1)^2 - X^2)$; $(\Delta^2 X^2)(0) = (\Delta(2X+1))(0) = \underline{2}$; $\underline{\underline{\varphi_2 = P_2 + 2P_3}}$

$(\Delta^0 X^3)(0) = \underline{0}$; $(\Delta^1 X^3)(0) = \underline{1}$ $(\Delta X^3 = (X+1)^3 - X^3)$; $\Delta^2 X^3(0) = (\Delta(3X^2+3X+1))(0) = \underline{6}$; $\Delta^3 X^3 = \Delta(6X+6) = \underline{6}$

$\underline{\underline{\varphi_3 = P_2 + 6P_3 + 6P_4}}$

$\underline{\underline{\varphi_1 = \frac{1}{2} X(X-1)}}$; $S_1(p) = \varphi_1(p+1) = \frac{(p+1)p}{2}$; $\underline{\underline{S_1(p) = \frac{p(p+1)}{2}}}$.

$\varphi_2 = \frac{1}{2} X(X-1) + \frac{2}{6} X(X-1)(X-2) = \frac{1}{6} X(X-1)(3 + 2X - 4)$; $S_2(p) = \varphi_2(p+1) = \frac{1}{6}(p+1)\,p\,(2(p+1)-1)$

$\underline{\underline{\varphi_2 = \frac{1}{6} X(X-1)(2X-1)}}$ $\underline{\underline{S_2(p) = \frac{p(p+1)(2p+1)}{6}}}$.

$\varphi_3 = \frac{1}{2} X(X-1) + \frac{6}{6} X(X-1)(X-2) + \frac{6}{24} X(X-1)(X-2)(X-3) = \frac{1}{4} X(X-1)\left[2 + 4(X-2) + (X-2)(X-3)\right]$

$\varphi_3 = \frac{1}{4} X(X-1)\left[2 + 4X - 8 + X^2 - 5X + 6\right] = \frac{1}{4} X(X-1)(X^2 - X) = \left[\frac{X(X-1)}{2}\right]^2$. $\varphi_3 = \left[\frac{X(X-1)}{2}\right]^2$

$\underline{\underline{S_3(p) = \left(\frac{p(p+1)}{2}\right)^2}}$.

__Remarque__.. Tout cela pour arriver à des résultats que l'on obtient en 3 lignes en 1ère A
Bravo, gégé tu es le plus fort !

PARTIE II Hors-d'œuvre N°2

__Q1.. Fonction génératrice d'une VAR..__

a.. X suit une loi de Bernoulli de paramètre λ.

$\forall t \in \mathbb{R}$, $G_X(t) = p(X=0)\,t^0 + p(X=1)\,t = (1-\lambda) + \lambda t$

$\underline{\underline{\forall t \in \mathbb{R},\ G_X(t) = \lambda t + (1-\lambda)}}$.

b.. X suit une loi binomiale de paramètre n et λ

$\forall t \in \mathbb{R}$, $G_X(t) = \sum_{k=0}^{n} p(X=k)\,t^k = \sum_{k=0}^{n} C_n^k \lambda^k (1-\lambda)^{n-k} t^k = (t\lambda + (1-\lambda))^n$

$\underline{\underline{\forall t \in \mathbb{R},\ G_X(t) = (\lambda t + (1-\lambda))^n}}$

Q2 .. Fonction génératrice d'une somme de variables aléatoires indépendantes..

a.. $Y_2 = X_1 + X_2$ et X_1 et X_2 sont indépendantes.

V1.. Soit $t \in \mathbb{R}$

$$G_{Y_2}(t) = \sum_{k=0}^{p_1+p_2} p(Y_2 = k)\, t^k = \sum_{k=0}^{p_1+p_2} \Big(\sum_{i=0}^{p_1} p(X_1 + X_2 = k \text{ et } X_1 = i) \Big) t^k$$

$$G_{Y_2}(t) = \sum_{k=0}^{p_1+p_2} \Big(\sum_{i=0}^{p_1} p(X_1 = i \text{ et } X_2 = k-i) \Big) t^k \underset{\uparrow}{=} \sum_{k=0} \Big(\sum_{i=\max(0,k-p_2)}^{i=\min(k,p_1)} p(X_1 = i)\, p(X_2 = k-i) \Big) t^k$$

Indépendance

De plus $G_{X_1}(t)\, G_{X_2}(t) = \Big(\sum_{k=0}^{p_1} p(X_1 = k)\, t^k\Big)\Big(\sum_{k=0}^{p_2} p(X_2 = k)\, t^k\Big) \underset{\uparrow}{=} \sum_{k=0}^{p_1+p_2} \Big(\sum_{i=\max(0,k-p_2)}^{\min(k,p_1)} p(X_1 = i)\, p(X_2 = k-i) \Big) t^k$

(produit polynômiale ... $(a_n)(b_n) = (c_n)$ avec $c_n = \sum_{j=0}^{n} a_j b_{n-j}$ ...)

Finalement : $\forall t \in \mathbb{R},\ G_{Y_2}(t) = G_{X_1}(t)\, G_{X_2}(t)$

Donc $G_{Y_2} = G_{X_1} G_{X_2}$.

ou $\underline{\underline{G_{Y_2} = G_{X_1+X_2} = G_{X_1} G_{X_2}}}$

V2.. Soit $t \in \mathbb{R}$. $G_{Y_2}(t) = \sum_{k=0}^{p_1+p_2} p(Y_2 = k)\, t^k \underset{\uparrow}{=} E(t^{Y_2}) = E(t^{X_1+X_2}) = E(t^{X_1} t^{X_2})$

Théorème de transfert

$G_{Y_2}(t) = E(t^{X_1} t^{X_2}) \underset{\uparrow}{=} E(t^{X_1})E(t^{X_2}) = G_{X_1}(t)\ G_{X_2}(t)$ ... cqfd

si X_1 et X_2 sont indépendantes t^{X_1} et t^{X_2} aussi ... théorème du programme ... "toute fonction... "

b.. Montrons par récurrence que : $\forall k \in [\![1,n]\!],\ G_{Y_k} = \prod_{i=1}^{k} G_{X_i}$

- c'est clair pour $k = 1$ $(Y_1 = X_1)$
- Supposons la propriété vraie pour $k \in [\![1, n-1]\!]$ et montrons la pour $k+1$.

$Y_{k+1} = Y_k + X_k$

$X_1, X_2, \dots, X_k, X_{k+1}$ sont indépendantes donc $X_1 + X_2 + \dots + X_k$ et X_{k+1} aussi (théorème du programme "toute fonction .... ") ; Y_k et X_{k+1} sont donc indépendantes.

Par conséquent d'après a) : $G_{Y_{k+1}} = G_{Y_k + X_{k+1}} = G_{Y_k} G_{X_{k+1}} \underset{\substack{\uparrow \\ HR}}{=} \Big(\prod_{i=1}^{k} G_{X_i}\Big) G_{X_{k+1}} = \prod_{i=1}^{k+1} G_{X_i}$

Ceci achève la récurrence.

Finalement $\underline{\underline{G_{Y_n} = G_{X_1+X_2+\dots+X_n} = \prod_{i=1}^{n} G_{X_i}}}$

Remarque.. On peut encore retrouver ce résultat avec les arguments de V2 a)

Application.. Supposons de plus que pour tout $i \in [\![1,n]\!]$, X_i suit une loi de Bernoulli de paramètre λ alors $Y_n \hookrightarrow \mathcal{B}(n, \lambda)$ (... indépendance). $G_{Y_n} = \prod_{i=1}^{n} G_{X_i} = \prod_{i=1}^{n} (\lambda t + (1-\lambda)) = (\lambda t + (1-\lambda))^n$. On retrouve ainsi Q1 b.

Q3.. Etude d'un cas particulier..

a) ⚠ $X_k(\Omega) = [\![0, k-1]\!]$ et non pas $[\![1, k]\!]$! Ce n'est donc prié de ranger ses belles formules ou de les translater !

Soit $k \in [\![1, n]\!]$.

$E(X_k) = \sum_{i=0}^{k-1} i\, p(X_k = i) = \frac{1}{k} \sum_{i=0}^{k-1} i = \frac{1}{k} \times \frac{(k-1)k}{2} = \frac{k-1}{2}$. $\underline{\underline{E(X_k) = \frac{k-1}{2}}}$.

$E(X_k^2) = \sum_{i=0}^{k-1} i^2 p(X_k = i) = \frac{1}{k} \sum_{i=0}^{k-1} i^2 = \frac{1}{k} \times \frac{(k-1)k(2k-1)}{6} = \frac{(k-1)(2k-1)}{6}$

$V(X_k) = \frac{(k-1)(2k-1)}{6} - \left(\frac{k-1}{2}\right)^2 = \frac{k-1}{12}[4k-2-3k+3] = \frac{k^2-1}{12}$; $\underline{\underline{V(X_k) = \frac{k^2-1}{12}}}$.

$E(Y_n) = \sum_{k=1}^{n} E(X_k) = \sum_{k=1}^{n} \frac{k-1}{2} = \frac{1}{2}\frac{(n-1)n}{2}$; $E(Y_n) = \frac{(n-1)n}{4}$

$V(Y_n) \underset{\text{indépendance}}{=} \sum_{k=1}^{n} V(X_k) = \sum_{k=1}^{n} \frac{k^2-1}{12} = \frac{1}{12}\sum_{k=1}^{n} k^2 - \frac{n}{12} = \frac{1}{12}\,\frac{n(n+1)(2n+1)}{6} - \frac{n}{12} = \frac{n}{72}\left[(n+1)(2n+1) - 6\right]$

$V(Y_n) = \frac{n}{72}(2n^2+3n-5) = \frac{n(n-1)(2n+5)}{72}$; $\underline{\underline{V(Y_n) = \frac{n(n-1)(2n+5)}{72}}}$.

b) $G_{n+1} = G_{Y_{n+1}} = G_{Y_n} G_{X_{n+1}}$

Soit $t \in \mathbb{R}$.

$G_{n+1}(t) = G_{Y_n}(t)\, G_{X_{n+1}}(t) = G_n(t)\left(\sum_{k=0}^{n} p(X_{n+1} = k)\, t^k\right) = \frac{1}{n+1} G_n(t) \sum_{k=0}^{n} t^k$

$G_{n+1}(t) = \frac{1}{n+1} \times \frac{1}{n!}\left(\sum_{k=0}^{\frac{n(n+1)}{2}} C_{n,k}\, t^k\right)\left(\sum_{k=0}^{n} t^k\right)$. Posons $\forall k \in [\![0, n]\!]$, $\beta_k = 1$

$G_{n+1}(t) = \frac{1}{(n+1)!} \sum_{k=0}^{\frac{n(n-1)}{2}+n} \left(\sum_{i=\max(k-n,0)}^{\min(\frac{n(n-1)}{2},k)} C_{n,i}\, \beta_{k-i}\right) t^k = \frac{1}{(n+1)!} \sum_{k=0}^{\frac{n(n+1)}{2}} \left(\sum_{i=\max(k-n,0)}^{\min(\frac{n(n-1)}{2},k)} C_{n,i}\right) t^k$

compte tenu du fait que $C_{n,i} = 0$ si $i < 0$ ou $i > \frac{n(n-1)}{2}$ nous pouvons encore

écrire $G_{n+1}(t) = \frac{1}{(n+1)!} \sum_{k=0}^{\frac{n(n+1)}{2}} \left(\sum_{i=k-n}^{k} C_{n,i}\right) t^k$ et ceci pour tout $t \in \mathbb{R}$

Par conséquent : $C_{n+1,k} = \sum_{i=k-n}^{k} C_{n,i}$ pour tout $k \in [\![0, \frac{n(n+1)}{2}]\!]$

Notons que ceci vaut encore pour $k < 0$ (car dans ce cas $C_{n+1,k} = 0$ et $C_{n,i} = 0$ pour i compris entre $k-n$ et k) et pour $k > \frac{n(n+1)}{2}$ (... $C_{n+1,k} = 0$ et $C_{n,i} = 0$ pour $i \in [\![k-n, k]\!]$ car $k-n > \frac{n(n-1)}{2}$).

Finalement : $\forall k \in \mathbb{Z}$, $C_{n+1,k} = \sum_{i=k-n}^{k} C_{n,i} = \sum_{j=0}^{n} C_{n,k-j}$

c] Montrons par récurrence (sur n) que : $\forall n \in \mathbb{N}^*$, $\forall k \in \mathbb{Z}$, $C_{n,k} \in \mathbb{N}$

→ Pour $n=1$. $\forall t \in \mathbb{R}$, $G_{X_1}(t) = 1$; $C_{1,k} = \begin{cases} 0 \text{ si } k \in \mathbb{Z}^* \\ 1 \text{ si } k = 0 \end{cases}$; la propriété est vraie.

→ Supposons la propriété vraie pour $n \in \mathbb{N}^*$ et montrons la pour $n+1$.

$\forall k \in \mathbb{Z}$, $C_{n+1,k} = \sum_{i=k-n}^{k} C_{n,i} \underset{\uparrow H.R.}{\in} \mathbb{N}$. Ceci achève la récurrence.

Soit $n \in \mathbb{N}^*$.

$$\sum_{k=0}^{n(n-1)/2} C_{n,k} = \sum_{k=0}^{\frac{n(n-1)}{2}} C_{n,k}\, 1^k = n!\, G_n(1) = n! \sum_{k=0}^{\frac{n(n-1)}{2}} p(Y_n = k) = n!$$

donc $\forall n \in \mathbb{N}^*$, $\sum_{k=0}^{\frac{n(n-1)}{2}} C_{n,k} = n!$

Utilité de la partie II ??

Voir à la fin un programme calculant les $C_{n,k}$

TRI BULLE — PARTIE III — Le plat de résistance

Q1.. Exemple de mise en œuvre de l'algorithme

Etat initial	1 2 3	2 3 1	3 1 2	1 3 2	3 2 1	2 1 3
Après le 1er appel		2 1 3	1 3 2	1 2 3	2 3 1	1 2 3
Après le 2ème appel		1 2 3	1 2 3		2 1 3	
Après le 3ème appel					1 2 3	

Nb. d'appels 0 2 2 1 3 1

Q2.. Action de l'algorithme..

a.. La valeur contenue dans $s[n]$ à l'issue du passage dans la boucle

for j:=1 to n-1 do

If s[j]>s[j+1] then Echange(s[i],s[j]) n'est autre que

$\max(s[1], s[2], .., s[n])$ soit : n. Pour démontrer ce résultat, assez évident, montrons par récurrence qu'à la fin du $j^{ème}$ passage dans la boucle, $s[j+1]$ contient $\max(s[1], s[2], .., s[j+1])$ et ceci pour tout $j \in [\![1, n-1]\!]$.

→ C'est clair pour $j=1$

Examinons le 1er passage. Si $s[1] > s[2]$ on échange les contenus de $s[1]$ et $s[2]$; $s[2]$ contient alors $\max(s[1], s[2])$. Si $s[1] < s[2]$ on ne fait rien et $s[2]$ contient encore $\max(s[1], s[2])$.

Supposons la propriété vraie pour $j \in [\![1, n-2]\!]$ et montrons la pour $j+1$

$s[j+1]$ contient $\max(s[1], s[2], \dots, s[j+1])$

Examinons le $j+1$ ième passage

1er cas.. $s[j+1] > s[j+2]$; on échange les contenus de $s[j+1]$ et $s[j+2]$

Le max de $(s[1], s[2], \dots, s[j], s[j+1], s[j+2])$ n'est autre que la valeur qui se trouvait au départ dans $s[j+1]$ et qui se trouve dans $s[j+2]$

Donc à la fin de cette étape : $s[j+2] = \max(s[1], \dots, s[j+1], s[j+2])$.

2ème cas.. $s[j+1] < s[j+2]$

Or $s[j+1] = \max s[1], \dots, s[j+1])$, donc $s[j+2] = \max(s[1], \dots, s[j+2])$

Ceci achève la récurrence et confirme le fait qu'à la fin de cette boucle $s[n]$ contient $\max(s[1], \dots, s[n])$ soit n.

b.. Il va maintenant de soi qu'à la fin de l'algorithme : $\forall k \in [\![1, n]\!]$, $s[k] = k$.

Montrons ce résultat par une récurrence

Plus précisément prouvons par une récurrence faible que pour tout $i \in [\![1, n-1]\!]$ à la fin du $i$ème passage dans la 1ère boucle $s[n-i+1]$ contient $n-i+1$

→ C'est clair pour $i = 1$ car $s[n]$ contient n à la fin du 1er passage dans la boucle externe d'après <u>a)</u>

→ Supposons la propriété vraie jusqu'à $i \in [\![1, n-2]\!]$ et montrons la pour $i+1$

(H) A la fin du $i$ème passage dans la boucle externe $s[n]$ contient n, $s[n-1]$ contient $n-1, \dots,$ $s([n-i+1])$ contient $n-i+1$; supposons donc le $i$ème passage effectué.

D'après <u>a)</u> le $i+1$ ème passage va mettre $\max(s[1], \dots, s[n-i])$ dans $s[n-i]$

Or $\underline{\underline{\{}} s[n-i+1], s[n-i+2], \dots, s[n-1], s[n] \underline{\underline{\}}} = \{n-i+1, n-i+2, \dots, n-1, n\}$

Donc $\underline{\underline{\{}} s[1], s[2], \dots, s[n-i] \underline{\underline{\}}} = \underline{\underline{\{}} 1, 2, \dots, n-i \underline{\underline{\}}}$

Par conséquent $\max(s[1], \dots, s[n-i]) = n-i$

Donc le $i+1$ ème passage met $n-i$ dans $s[n-i]$ ou $n-(i+1)-1$ dans $s[n-(i+1)+1]$;

ceci achève la récurrence.

Nous avons donc prouvé qu'à la fin de l'algorithme pour tout $i \in [\![1, n-1]\!]$, $s[n-i+1]$ contient $n-i+1$ <u>ou</u> pour tout $k \in [\![2, n]\!]$, $s[k]$ contient k

Reste à dire que le seul élément que peut contenir $s[1]$ est 1 !

Finalement à la fin de l'algorithme $s[k]$ contient k pour tout $k \in [\![1, n]\!]$.

Le tableau a donc été trié dans l'ordre croissant.

<u>Q3.. Complexité de l'algorithme</u>

a.. Le $i$ème passage dans la boucle externe voit s'effectuer $n-i$ comparaisons et ceci pour tout $i \in [\![1, n-1]\!]$.

Le nombre total de comparaisons effectuées au cours de l'algorithme est $\sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2}$

b) Le nombre minimal d'appels à la procédure échange est 0 ; c'est le cas lorsque l'on affecte initialement à $(s[1], s[2], \dots, s[n])$: $(1, 2, \dots, n)$.

Le nombre maximal d'appels à la procédure échange est $\frac{n(n-1)}{2}$, c'est le cas lorsque l'on affecte initialement à $(s[1], s[2], \dots, s[n])$: $(n, n-1, \dots, 2, 1)$. Dans ce cas chaque passage dans la boucle interne où chaque comparaison provoque un échange.

Remarque.. Les résultats ci-dessus peuvent être démontrés à l'aide de récurrences comme dans Q2. Cela ne semble pas très opportun ici vu la longueur du texte.

Q4.. Nombre d'appels de la procédure échanger..

a.. Supposons $s[j] > s[j+1]$ et échangeons les valeurs de $s[j]$ et $s[j+1]$.

$s[1], \dots, s[j-1], s[j+2], \dots, s[n]$ ne sont pas modifiés.

On ne rajoute ni n'enlève aucune inversion associée aux couples (i, k) tels que $1 \leq i < k \leq j$ ou tels que : $1 \leq i < k$ et $k \geq j+2$ (les éléments avant l'indice k sont "globalement" inchangés pour tout $k \in [\![1, n]\!] - \{j+1\}$)

Reste à examiner les inversions "$(i, j+1)$" avec $i < j+1$

Si $i < j$ elles demeurent

Si $i = j$ l'inversion initiale ($s[j] > s[j+1]$ avant l'échange) disparaît (on a $s[j] < s[j+1]$ après l'échange).

Finalement si $s[j] > s[j+1]$ l'échange des valeurs de $s[j]$ et $s[j+1]$ diminue de 1 le nombre d'inversions du tableau.

Trier s dans l'ordre croissant c'est rendre nul le nombre d'inversions du tableau ; or chaque appel à la procédure Echanger diminue de 1 le nombre d'inversions du tableau. Par conséquent le nombre d'appels de la procédure Echanger est le nombre $V_n(s)$ d'inversions du tableau.

b..

	(1,2,3)	(2,3,1)	(3,1,2)	(1,3,2)	(3,2,1)	(2,1,3)
$U_1(s)$	0	0	0	0	0	0
$U_2(s)$	0	0	1	0	1	1
$U_3(s)$	0	2	1	1	2	0
$V_3(s)$	0	2	2	1	3	1

c)

```
WRITELN ('U(1) vaut 0'); INVERSENOI := 0;
FOR I := 2 TO N DO
BEGIN
 GRANDFOU := 0; A := S[I];
 FOR J := 1 TO I-1 DO
 IF S[J] > A THEN GRANDFOU := GRANDFOU + 1;
 WRITELN ('U(', I, ') vaut ', GRANDFOU);
 INVERSENOI := INVERSENOI + GRANDFOU;
END;
WRITELN; WRITELN ('le nombre d''inversion est : ', INVERSENOI);
```

Voir à la fin un programme complet.

Intéressant car la comparaison de $s[J]$ avec A est plus rapide que celle de $s[J]$ avec $s[I]$ (l'adresse de A est moins complexe que celle de $s[I]$)

Q5.. Bijectivité de l'application $s \to (U_1(s), U_2(s), \dots, U_n(s))$.

a.. Soit $s \in S_n$.

Si $k \in [\![2, n]\!]$, $U_k(s) = \text{card}\{i \in [\![1, k-1]\!] \mid s[i] > s[k]\} \leq \text{card}\,[\![1, k-1]\!] = k-1 < k$

Si $k = 1$: $U_1(s) = 0 < 1$

Donc $\forall k \in [\![1, n]\!]$, $U_k(s) \in [\![1, k-1]\!]$ ou $\forall k \in [\![1, n]\!]$, $U_k(s)$ est un entier strictement inférieur à k.

Donc $\forall s \in S_n$, $(U_1(s), U_2(s), \dots, U_n(s)) \in \Omega_n$.

Remarque.. $\Omega_n = [\![0, 0]\!] \times [\![0, 1]\!] \times \dots \times [\![0, n-1]\!]$, par conséquent $\text{card}\,\Omega_n = 1 \times 2 \times \dots \times n = n!$

Donc $\underline{\underline{\text{card}\,\Omega_n = \text{card}\,S_n = n!}}$

b.. Soit $s \in S_n$.

u_n est le nombre d'éléments de la liste $(s[1], \dots, s[n-1])$ strictement supérieurs à $s[n]$
$n-1-u_n$ est donc le nombre d'éléments de la liste $(s[1], \dots, s[n-1])$ donc de la liste $(1, 2, \dots, n)$ strictement inférieurs à $s[n]$; par conséquent $s[n]$ occupe le rang $n-1-u_n+1$ de cette dernière liste ; ceci donne donc $\underline{\underline{s[n] = n - u_n}}$.

Notons L_{n-1} la liste $(1, 2, \dots, n)$ privée de l'élément $s[n]$; les éléments de L_{n-1} sont encore les éléments de $(s[1], s[2], \dots, s[n-1])$.
u_{n-1} compte le nombre d'éléments de la liste $(s[1], \dots, s[n-2])$ strictement supérieurs à $s[n-1]$
$(n-2)-u_{n-1}$ compte le nombre d'éléments de la liste $(s[1], \dots, s[n-2])$ donc de la liste L_{n-1} strictement inférieurs à $s[n-1]$; par conséquent $s[n-1]$ occupe le rang $(n-2)-u_{n-1}+1$ dans la liste L_{n-1}. $s[n-1]$ est donc le $(n-1)-u_{n-1}$ ième élément de la liste L_{n-1}

Une récurrence élémentaire prouve alors que pour tout $k \in [\![1, n]\!]$, $s[k]$ est le $k-u_k$ ième élément de la liste L_k obtenue à partir de $(1, 2, \dots, n)$ en retirant les éléments $s[k+1], s[k+2], \dots, s[n]$ (... à un abus près : $k = n$ .. dans ce cas on ne retire rien)

De manière formelle si pour tout $p \in [\![0, n]\!]$ et pour toute partie A de $[\![1, n]\!]$ ayant au moins p éléments on s'autorise à noter $r_p(A)$ le $p^{\text{ième}}$ élément de la liste des éléments de A ordonnés de manière croissante on a :

$$\forall k \in [\![1, n]\!], s[k] = r_{k-u_k}\left(\{1, 2, \dots, n\} - \{s[k+1], s[k+2], \dots, s[n]\}\right).$$

Remarque .. $(v_1(s), v_2(s), \dots, v_n(s))$ est la table des inversions du tableau s (... ou de la permutation) Comme nous allons le voir à travers la bijection suivante, elle détermine entièrement s.

c .. Considérons l'application U de S_n dans $\mathcal{R}_n$ qui à $s \in S_n$ associe $U(s) = (v_1(s), v_2(s), \dots, v_n(s))$ (Notons que le texte, si je ne m'abuse ne définit pas $U(s)$)

Montrons que U est une bijection de S_n sur $\mathcal{R}_n$. Comme $\operatorname{card} S_n = n! = \operatorname{card} \mathcal{R}_n < +\infty$ il suffit de prouver que U est injective. Soient s et t deux éléments de S_n tels que $U(s) = U(t)$ montrons que $s = t$. Pour cela prouvons par une récurrence faible et descendante que : $\forall k \in [\![1, n]\!]$, $s[k] \overset{*}{=} t[k]$. Posons $\forall k \in [\![1, n]\!]$, $u_k = v_k(s)$ et $u'_k = v_k(t)$.

$U(s) = U(t)$ donne donc $\forall k \in [\![1, n]\!], u_k = u'_k$

→ L'égalité * vaut pour $k = n$ car :

$s[n] = r_{n-u_n}(\{1, 2, \dots, n\}) = r_{n-u'_n}(\{1, 2, \dots, n\}) = t[n]$.

→ Supposons que * soit vraie pour tout $i \in [\![k, n]\!]$ ($k \in [\![2, n]\!]$) et montrons la alors pour $k-1$.

L'hypothèse de récurrence indique que $\forall i \in [\![k, n]\!]$, $s[i] = t[i]$; prouvons alors que : $s[k-1] = t[k-1]$.

$$s[k-1] = r_{(k-1)-u_{k-1}}\left(\{1, 2, \dots, n\} - \{s[k], \dots, s[n]\}\right) \underset{\substack{\uparrow \\ \text{HR et } u_{k-1} = u'_{k-1}}}{=} r_{k-1-u'_{k-1}}\left(\{1, 2, \dots, n\} - \{t[k], \dots, t[n]\}\right) = t[k-1].$$

ce qui achève la récurrence ; U est donc injective et par conséquent bijective car $\operatorname{card} S_n = \operatorname{card} \mathcal{R}_n < +\infty$.

Remarque .. Il importe de bien savoir passer d'un tableau ou d'une permutation à sa table d'inversion et réciproquement.

Le 1er passage est illustré par Q4 .. Illustrons le passage inverse.

Prenons $n=6$ et cherchons le tableau dont la table est $(0,1,1,2,3,2)$

$U_6(\sigma)=2$; $\sigma[6]$ est $6-2$ ième élément de la liste $L_6=(1,2,3,4,5,6)$; $\sigma[6]=4$

$U_5(\sigma)=3$; $\sigma[5]$ " $5-3$ ième " " " $L_5=(1,2,3,5,6)$; $\sigma[5]=2$

$U_4(\sigma)=2$; $\sigma[4]$ " $4-2$ ième " " " $L_4=(1,3,5,6)$; $\sigma[4]=3$

$U_3(\sigma)=1$; $\sigma[3]$ " $3-1$ ième " " " $L_3=(1,5,6)$; $\sigma[3]=5$

$U_2(\sigma)=1$; $\sigma[2]$ " $2-1$ ième " " " $L_2=(1,6)$; $\sigma[2]=1$

Et donc $\sigma[1]=6$

$\sigma=(6,1,5,3,2,4)$

Exercice de contrôle.. Prouver directement la surjectivité.. Ecrire un programme permettant de passer de la table d'inversion au tableau.

$k\in[\![1,n]\!]$ et $u_k\in[\![0,k-1]\!]$.

U étant bijective, le nombre d'éléments $\sigma\in S_n$ tels que : $U_k(\sigma)=u_k$ est le nombre d'éléments $(v_1,v_2,\dots,v_n)\in\Omega_n=[\![0,0]\!]\times[\![0,1]\!]\times\dots\times[\![0,n-1]\!]$ tels que : $v_k=u_k$. Ce nombre n'est autre que le cardinal de : $[\![0,0]\!]\times[\![0,1]\!]\times\dots\times[\![0,k-2]\!]\times[\![0,k]\!]\times\dots\times[\![0,n-1]\!]$ soit $\dfrac{n!}{k}$

(v_k est fixé mais v_1 peut être choisi dans $[\![0,0]\!]$, v_2 dans $[\![0,1]\!]$, ..., v_{k-1} dans $[\![0,k-2]\!]$, v_{k+1} dans $[\![0,k]\!]$, ..., v_n dans $[\![0,n-1]\!]$).

Q6.. Lois des variables aléatoires $U_1,U_2,\dots,U_n$.

a) Soit $(u_1,u_2,\dots,u_n)\in\Omega_n$. $\exists!\,\sigma\in S_n$, $U_1(\sigma)=u_1$, $U_2(\sigma)=u_2$, ..., $U_n(\sigma)=u_n$

Par conséquent $p(U_1=u_1,U_2=u_2,\dots,U_n=u_n)=\dfrac{1}{n!}$ (card $S_n=n!$)

b) $k\in[\![1,n]\!]$ et $u_k\in[\![0,k-1]\!]$.

$$p(U_k=u_k)=p(\{\sigma\in S_n\mid U_k(\sigma)=u_k\})=\frac{\mathrm{card}\{\sigma\in S_n\mid U_k(\sigma)=u_k\}}{n!}=\frac{n!/k}{n!}=\frac{1}{k}.$$

$U_k(S_n)=[\![0,k-1]\!]$ et $\forall u_k\in[\![0,k-1]\!]$, $p(U_k=u_k)=\dfrac{1}{k}$.

U_k suit donc une loi uniforme sur $[\![0,k-1]\!]$.

$\forall k\in[\![1,n]\!]$, $U_k(S_n)=[\![0,k-1]\!]$ et $\prod_{k=1}^{n}[\![0,k-1]\!]=\Omega_n$

$$\forall(u_1,u_2,\dots,u_n)\in\Omega_n,\ p(U_1=u_1,U_2=u_2,\dots,U_n=u_n)\underset{a)}{=}\frac{1}{n!}=1\times\frac12\times\frac13\times\dots\times\frac1n=p(U_1=u_1)p(U_2=u_2)\cdots p(U_n=u_n)$$

Ceci signifie donc que $U_1,U_2,\dots,U_n$ sont mutuellement indépendantes.

c) Le nombre d'appel de la procédure Echanger au cours de l'algorithme est $V_n=U_1+U_2+\dots+U_n$

$E(V_n)=\dfrac{n(n-1)}{4}$ et $V(V_n)=\dfrac{n(n-1)(2n+5)}{72}$ d'après II § 3 a). $E(V_n)\sim\dfrac{n^2}{4}$ et $V(V_n)\sim\dfrac{n^3}{36}$

La place me manque pour connaître mais on se retrouvera gigi !

Programme donnant les $C_{n,k}$

```
Program essec92p;
uses crt;
const dimmax=45;
var i,k,n,stop,deg,min,s:integer;c:array[0..dimmax] of integer;
Begin
clrscr;
Write('Donnez la valeur de n (n entre 1 et 10). n=');readln(stop);
deg:=stop*(stop-1);deg:=deg div 2;
for i:=0 to deg do c[i]:=0;
c[0]:=1;
Writeln(c[0]:4);
for n:=2 to stop do
    begin
    deg:=n*(n-1) div 2;
    for k:=deg downto 0 do
        begin
        s:=0;min:=k+1-n;if min<0 then min:=0;
        for i:= min to k do s:=s+c[i];
        Write(s:4);
        c[k]:=s
        end;
    writeln
    end;
end.
```

```
Donnez la valeur de n (n entre 1 et 10). n=5
   1
   1   1
   1   2   2   1
   1   3   5   6   5   3   1
   1   4   9  15  20  22  20  15   9   4   1
```

```
Donnez la valeur de n (n entre 1 et 10). n=7
   1
   1   1
   1   2   2   1
   1   3   5   6   5   3   1
   1   4   9  15  20  22  20  15   9   4   1
   1   5  14  29  49  71  90 101 101  90  71  49  29  14   5   1
   1   6  20  49  98 169 259 359 455 531 573 573 531 455 359 259 169  98  49  2(
   6   1
```

Programme donnant la table d'inversions

```
program essec92T;

uses crt;
const dimmax=20;
var inversemoi,grandfou,i,j,n,a:integer;s:array[1..dimmax] of integer;

begin
{Entrée du tableau}
clrscr;
Writeln('Entrée du tableau s');
write('Donnez la dimension n du tableau. n=');readln(n);
writeln('Donnez : ');
for i:=1 to n do
    begin
    write('   s[',i,']; s[',i,']=');read(s[i]);
    end;
{table des inversions}
writeln;
Write('U1(s)=0   ');inversemoi:=0;
for i:=2 to n do
    begin
    grandfou:=0;a:=s[i];
    for j:=1 to i-1 do
    if s[j]>a then grandfou:=grandfou+1;
    write('U',i,'(s)=',grandfou,'   ');
    inversemoi:=inversemoi+grandfou;
    end;
    writeln;writeln('Le nombre d''inversions est : ',inversemoi);
end.
```

```
Entrée du tableau s
Donnez la dimension n du tableau. n=8
Donnez :
   s[1]; s[1]=6
   s[2]; s[2]=5
   s[3]; s[3]=7
   s[4]; s[4]=3
   s[5]; s[5]=8
   s[6]; s[6]=2
   s[7]; s[7]=1
   s[8]; s[8]=4

U1(s)=0   U2(s)=1   U3(s)=0   U4(s)=3   U5(s)=0   U6(s)=5   U7(s)=6   U8(s)=4

Le nombre d'inversions est : 19
```

Programme triant un tableau avec la méthode du tri bulle

```
program essec92B;

uses crt;
const dimMax=30;LongMax=10;
type mot=string[LongMax];
type tab=array[1..dimMax] of mot;

var i,j,n,l:integer;s:tab;
procedure echanger(var chapi,chapeau:mot);
var zebulon:mot;
begin
zebulon:=chapeau;chapeau:=chapi;chapi:=zebulon;
end;

procedure entre(var fin:integer;var liste:tab);
var k:integer;
begin
writeln('Entrée du tableau s.');
Write('Donnez la dimension n du tableau s. n=');readln(fin);
Writeln('Donnez:');
for k:=1 to fin do
    begin
    write('s[',k,']; s[',k,']=');readln(liste[k]);
    end;
end;

begin
clrscr;
entre(n,s);
for i:=1 to n-1 do
    begin
    writeln;
    for j:=1 to n-i do
    if s[j]>s[j+1] then echanger(s[j],s[j+1]);
    for l:=1 to n do
        begin
        write(s[l],'  ');
        end;
    end;
end.
```

```
Entrée du tableau s.
Donnez la dimension n du tableau s. n=9
Donnez:
s[1]; s[1]=LETESSIER
s[2]; s[2]=AGANIAN
s[3]; s[3]=FLEURY
s[4]; s[4]=KAHN
s[5]; s[5]=THEVENOT
s[6]; s[6]=GUEGAN
s[7]; s[7]=CORREIA
s[8]; s[8]=JOST
s[9]; s[9]=LEHMAN

AGANIAN  FLEURY  KAHN  LETESSIER  GUEGAN  CORREIA  JOST  LEHMAN  THEVENOT
AGANIAN  FLEURY  KAHN  GUEGAN  CORREIA  JOST  LEHMAN  LETESSIER  THEVENOT
AGANIAN  FLEURY  GUEGAN  CORREIA  JOST  KAHN  LEHMAN  LETESSIER  THEVENOT
AGANIAN  FLEURY  CORREIA  GUEGAN  JOST  KAHN  LEHMAN  LETESSIER  THEVENOT
AGANIAN  CORREIA  FLEURY  GUEGAN  JOST  KAHN  LEHMAN  LETESSIER  THEVENOT
AGANIAN  CORREIA  FLEURY  GUEGAN  JOST  KAHN  LEHMAN  LETESSIER  THEVENOT
AGANIAN  CORREIA  FLEURY  GUEGAN  JOST  KAHN  LEHMAN  LETESSIER  THEVENOT
AGANIAN  CORREIA  FLEURY  GUEGAN  JOST  KAHN  LEHMAN  LETESSIER  THEVENOT
```

```
Entrée du tableau s.
Donnez la dimension n du tableau s. n=8
Donnez:
s[1]; s[1]=SAUVEL
s[2]; s[2]=POTTIER
s[3]; s[3]=DUPLA
s[4]; s[4]=STEINMANN
s[5]; s[5]=DUSSEAUX
s[6]; s[6]=PIGEASSOU
s[7]; s[7]=DARBOIS
s[8]; s[8]=BERGES

POTTIER  DUPLA  SAUVEL  DUSSEAUX  PIGEASSOU  DARBOIS  BERGES  STEINMANN
DUPLA  POTTIER  DUSSEAUX  PIGEASSOU  DARBOIS  BERGES  SAUVEL  STEINMANN
DUPLA  DUSSEAUX  PIGEASSOU  DARBOIS  BERGES  POTTIER  SAUVEL  STEINMANN
DUPLA  DUSSEAUX  DARBOIS  BERGES  PIGEASSOU  POTTIER  SAUVEL  STEINMANN
DUPLA  DARBOIS  BERGES  DUSSEAUX  PIGEASSOU  POTTIER  SAUVEL  STEINMANN
DARBOIS  BERGES  DUPLA  DUSSEAUX  PIGEASSOU  POTTIER  SAUVEL  STEINMANN
BERGES  DARBOIS  DUPLA  DUSSEAUX  PIGEASSOU  POTTIER  SAUVEL  STEINMANN
```