1ère S1 **Devoir pour le vendredi 25 janvier 2013**

I. Formules sommatoires

Le but de cet exercice est d'étudier un procédé classique de simplification appelé simplification par « télescopage » ou « en cascade » afin d'établir des formules sommatoires.

Partie A

Pour tout entier naturel $n \geqslant 1$, on pose : $S_n = 1+2+...+n$ (ou $S_n = \sum_{k=1}^{k=n} k$).

1°) On considère le polynôme $P(x) = \dfrac{x(x-1)}{2}$.

Vérifier que, pour tout réel x, on a : $P(x+1) - P(x) = x$.

2°) On écrit l'égalité $P(x+1) - P(x) = x$ pour $x \in \{1, 2, ..., n\}$ (l'entier naturel $n \geqslant 1$ étant fixé) et on fait la somme membre à membre comme dans le cadre ci-dessous (les signes « = » bien en dessous les uns des autres).

$$\boxed{\begin{array}{c} P(2)-P(1)=1 \\ P(3)-P(2)=2 \\ \dots\dots\dots\dots\dots\dots \\ P(n)-P(n-1)=n-1 \\ P(n+1)-P(n)=n \end{array}}$$

Recopier ce cadre, tirer un trait en dessous et barrer en diagonale les termes qui s'annulent dans les membres de gauche.

En déduire une expression simplifiée de S_n en fonction de n.

Partie B

Pour tout entier naturel $n \geqslant 1$, on pose : $S'_n = 1^2 + 2^2 + ... + n^2$ (ou $S'_n = \sum_{k=1}^{k=n} k^2$).

Reprendre la démarche de la **Partie A**, en considérant le polynôme $Q(x) = \dfrac{x(x-1)(2x-1)}{6}$ afin de donner une expression simplifiée de S'_n en fonction de n.

Partie C (facultative)

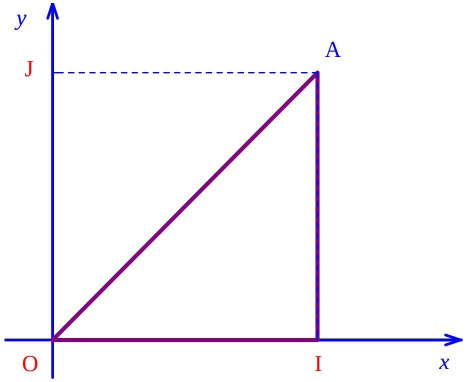
En utilisant une démarche analogue à celle des parties précédentes, établir une expression simplifiée de

$S''_n = 1^3 + 2^3 + ... + n^3$ (ou $S''_n = \sum_{k=1}^{k=n} k^3$).

II. Algorithmes de hachurage d'une zone de plan

Le plan est muni d'un repère orthonormé (O, I, J).

On souhaite réaliser des algorithmes permettant de hachurer de diverses manières l'intérieur du triangle OIA où A est le point de coordonnées (1 ; 1). Les hachures sont régulièrement espacées.



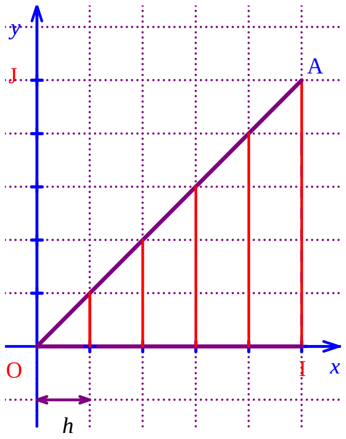
Écrire trois algorithmes séparés* demandant à l'utilisateur une valeur de n (nombre de hachures, qui sera un entier naturel supérieur ou égal à 1) afin de réaliser des figures semblables à celles présentées ci-dessous (dans le cas particulier où $n = 5$).

Au début de l'algorithme, on écrira les instructions permettant de tracer le triangle OIA.

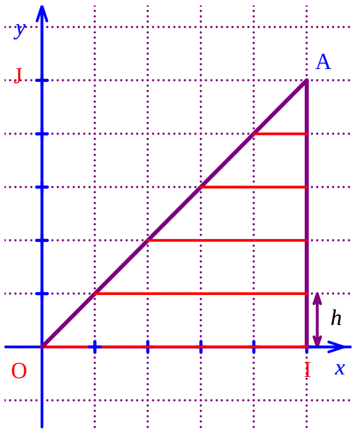
On emploiera une variable h.

On rappelle que I est le point unité sur l'axe des abscisses et donc que le segment [OI] représente l'intervalle [0 ; 1] sur l'axe des abscisses.

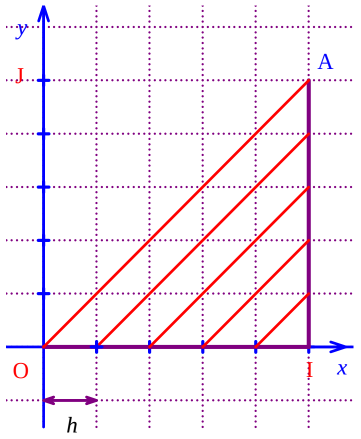
Chaque fois, l'un des côtés du triangle sera compté dans les hachures.



Hachurage vertical



Hachurage horizontal



Hachurage par des segments de coefficient directeur 1*

Réaliser les programmes correspondants sur calculatrice ou sur un logiciel de programmation.

Exécuter les programmes en prenant des valeurs de n de plus en plus grandes afin d'observer des hachures de plus en plus fines.

* On peut faire un algorithme commun pour les trois hachurages, en introduisant une condition OUI/NON pour choisir quel "type" de hachurage effectuer.

Conseils

Ne rien écrire sur l'énoncé.

I.

Les sommes considérées dans tout le devoir ont déjà été rencontrées dans le devoir précédent.

La somme de la **Partie A** a déjà été rencontrée dans la narration de recherche sur les segments.

Écrire le cadre en haut de la 2e page de la copie (pour avoir la place de montrer les calculs qui vont suivre).

Penser à quantifier les égalités.

2°) **Écrire le cadre en haut de la 3e page de la copie.**

Bien suivre l'énoncé (sans se laisser perturber par le fait que l'on veut calculer la somme des carrés).

Écrire au moins les deux premières lignes (pour k = 1 et k = 2). Mettre des petits points.
Écrire la dernière égalité pour $k = n$.

$$\boxed{\begin{array}{l} (1+1)^3 = \ldots \\ (2+1)^3 = \ldots \\ \\ \ldots \\ \\ (n+1)^3 = \ldots \end{array}}$$

On peut éventuellement utiliser des **outils de calcul formel**.

Il existe une commande permettant de donner une formule explicite pour une telle somme.

II. Essayer pour des valeurs de n assez grandes 20, 30, 50…

Facultatif :

Écrire et programmer des algorithmes permettant d'autres systèmes de hachures : hachures par des segments de coefficient directeur – 1*, doubles hachures etc.

* Le mot coefficient directeur est en général réservé aux droites. On l'emploie ici abusivement pour un segment par souci de simplification.

Corrigé du DM pour le 25-1-2013

I. Formules sommatoires

Partie A

$S_n = 1+2+...+n$ (ou $S_n = \sum_{k=1}^{k=n} k$) avec $n \geqslant 1$

1°) $P(x) = \dfrac{x(x-1)}{2}$

Vérifions que $\forall x \in \mathbb{R}$ $P(x+1) - P(x) = x$.

$\forall x \in \mathbb{R}$ $P(x+1) = \dfrac{(x+1)x}{2}$

$$\forall x \in \mathbb{R} \quad P(x+1) - P(x) = \frac{(x+1)x}{2} - \frac{x(x-1)}{2}$$
$$= \frac{x^2+x}{2} - \frac{x^2-x}{2}$$
$$= \frac{x^2+x-x^2+x}{2}$$
$$= x$$

2°) **Déduisons-en une expression de S_n en fonction de n.**

On écrit l'égalité $P(x+1) - P(x) = x$ pour $x \in \{1, 2, ..., n\}$ et on fait la somme membre à membre comme dans le cadre ci-dessous.

$$\boxed{\begin{array}{c} \cancel{P(2)} - P(1) = 1 \\ P(3) - \cancel{P(2)} = 2 \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ \cancel{P(n)} - P(n-1) = n-1 \\ P(n+1) - \cancel{P(n)} = n \end{array}}$$

On obtient : $P(n+1) - P(1) = 1 + 2 + 3 + \ldots + (n-1) + n$

Or $P(1) = \dfrac{1\times(1-1)}{2} = \dfrac{1\times 0}{2} = 0$ et $P(n+1) = \dfrac{(n+1)n}{2}$

D'où $S_n = \dfrac{(n+1)n}{2}$.

Partie B

$S'_n = 1^2 + 2^2 + ... + n^2$ (ou $S'_n = \sum_{k=1}^{k=n} k^2$)

$Q(x) = \dfrac{x(x-1)(2x-1)}{6}$

Vérifions que $\forall x \in \mathbb{R}$ $Q(x+1) - Q(x) = x^2$.

$\forall x \in \mathbb{R}$ $Q(x+1) = \dfrac{(x+1)x(2x+1)}{2}$

$$\forall x \in \mathbb{R} \quad Q(x+1) - Q(x) = \frac{(x+1)x(2x+1)}{6} - \frac{x(x-1)(2x-1)}{6}$$
$$= \frac{x(x+1)(2x+1)}{6} - \frac{x(x-1)(2x-1)}{6}$$
$$= \frac{x\left[2x^2+3x+1-\left(2x^2-3x+1\right)\right]}{6}$$
$$= \frac{x\left(\cancel{2x^2}+3x\cancel{+1}\cancel{-2x^2}+3x\cancel{-1}\right)}{6}$$
$$= \frac{6x\times x}{6}$$
$$= x^2$$

Exprimons S'_n en fonction de n.

On écrit l'égalité $Q(x+1) - Q(x) = x^2$ pour $x \in \{1, 2, ..., n\}$ et on fait la somme membre à membre comme dans le cadre ci-dessous.

$$\boxed{\begin{array}{c} \cancel{Q(2)} - Q(1) = 1^2 \\ Q(3) - \cancel{Q(2)} = 2^2 \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ \cancel{Q(n)} - Q(n-1) = (n-1)^2 \\ Q(n+1) - \cancel{Q(n)} = n^2 \end{array}}$$

On obtient : $Q(n+1) - Q(1) = S'_n$

Or $Q(1) = 0$ donc $Q(n+1) = S'_n$

On en déduit que $S'_n = \dfrac{n(n+1)(2n+1)}{6}$.

Partie C

$S''_n = 1^3 + 2^3 + ... + n^3$ (ou $S''_n = \sum_{k=1}^{k=n} k^3$).

Pour adapter la méthode des parties précédentes, la difficulté est de trouver un polynôme R qui vérifie
$\forall x \in \mathbb{R} \quad R(x+1) - R(x) = x^3$.

La difficulté consiste à trouver un tel polynôme.

On pouvait s'inspirer du devoir précédent (formules sommatoires par la méthode de Pascal) ou bien utiliser un logiciel de calcul formel.

On peut prendre le polynôme $R(x) = \left[P(x)\right]^2$ c'est-à-dire $R(x) = \left[\frac{x(x-1)}{2}\right]^2$ ou encore $R(x) = \frac{x^2(x-1)^2}{4}$.

On peut alors vérifier aisément que $\forall x \in \mathbb{R}$ $R(x+1) - R(x) = x^3$ (soit « à la main », car les calculs sont simples, soit à l'aide d'un logiciel de calcul formel) :

$$\begin{aligned}
\forall x \in \mathbb{R} \quad R(x+1) - R(x) &= \frac{x^2(x+1)^2 - x^2(x-1)^2}{4} \\
&= \frac{x^2\left[(x+1)^2 - (x-1)^2\right]}{4} \\
&= \frac{x^2(x+1-x+1)(x+1+x-1)}{4} \\
&= \frac{x^2 \times 2 \times 2x}{4} \\
&= \frac{4x^3}{4} \\
&= x^3
\end{aligned}$$

En procédant de la manière que dans les parties précédentes, $R(n+1) - R(1) = S''_n$.

Or $R(1) = 0$ donc $R(n+1) = S''_n$.

On en déduit que $S''_n = \frac{n^2(n+1)^2}{4}$.

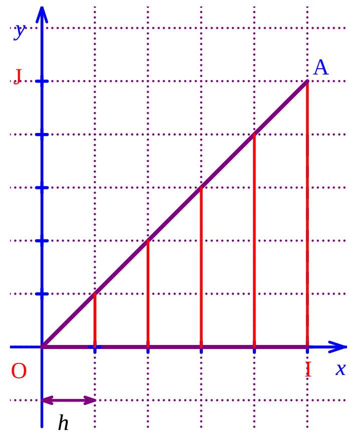
On retrouve $S''_n = (S_n)^2$.

Information :

Au XVI^e siècle, Johann Faulhaber donne les formules de plusieurs calculs de la somme des puissances des entiers ou formules sommatoires (voir renseignements sur Internet, par exemple sur Wikipedia).

II. Algorithme de construction géométrique (« hachuration »)

Il s'agit de construction itérative.

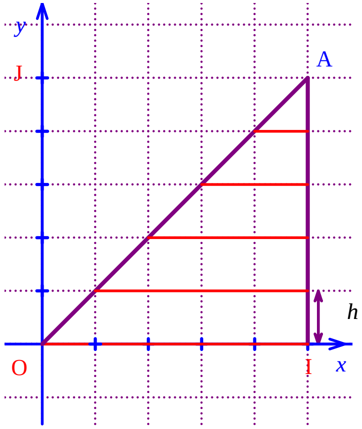


Variables :
n entier, supérieur ou égal à 1
h réel

Entrée :
Saisir n

Traitement et sortie :
Tracer le segment joignant les points de coordonnée (0 ; 0) et (1 ; 1)
Tracer le segment joignant les points de coordonnée (1 ; 1) et (1 ; 0)
Tracer le segment joignant les points de coordonnée (0 ; 0) et (1 ; 0)

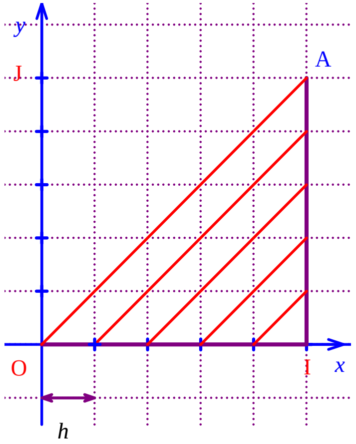
Pour h allant de 0 à 1 avec un pas de $\frac{1}{n}$ **Faire**
  Tracer le segment joignant les points de coordonnée (h ; h) et (h ; 0)
FinPour



Variables :
n entier supérieur ou égal à 1
h réel

Entrée :
Saisir *n*

Traitement et sortie :
Tracer le segment joignant les points de coordonnée (0 ; 0) et (1 ; 1)
Tracer le segment joignant les points de coordonnée (1 ; 1) et (1 ; 0)
Tracer le segment joignant les points de coordonnée (0 ; 0) et (1 ; 0)
Pour *h* allant de 0 à 1 avec un pas de $\frac{1}{n}$ **Faire**
 Tracer le segment joignant les points de coordonnée $(1 - h\ ;\ 1 - h)$ et $(1\ ;\ 1 - h)$
FinPour



Variables :
n entier supérieur ou égal à 1
h réel

Entrée :
Saisir *n*

Traitement et sortie :
Tracer le segment joignant les points de coordonnée (0 ; 0) et (1 ; 1)
Tracer le segment joignant les points de coordonnée (1 ; 1) et (1 ; 0)
Tracer le segment joignant les points de coordonnée (0 ; 0) et (1 ; 0)
Pour *h* allant de 0 à 1 avec un pas de $\frac{1}{n}$ **Faire**
 Tracer le segment joignant les points de coordonnée $(1 - h\ ;\ 0)$ et $(1\ ;\ h)$
FinPour

On aurait aussi pu réaliser un système de doubles hachures superposées (horizontales et verticales par exemple).

Algorithmes d'Alexandre Contassot

Algorithme 1

Entrée :
Saisir N

Initialisations* :
Tracer le segment reliant les points de coordonnée (0 ; 0) et (1 ; 1)
Tracer le segment reliant les points de coordonnée (0 ; 0) et (1 ; 0)
Tracer le segment reliant les points de coordonnée (1 ; 0) et (1 ; 1)
H prend la valeur $\frac{1}{N}$
P prend la valeur 1

Traitement et sortie :
Pour *i* allant de 1 à N **Faire**
  Tracer le segment reliant les points de coordonnée (P ; 0) et (P ; P)
  P prend la valeur P – H
FinPour

```
: Eff Dessin
: Ligne( 0,0,1,1)
: Ligne(0,0,1,0)
: Ligne(1,0,1,1)
: Prompt N
: 1/N→H
: 1→P
: For(I,1,N)
: Ligne(P,0 ,P,P)
: P – H →P
: End
```

* Le terme d'initialisation est discutable pour les trois premières instructions.

Algorithme 2

Entrée :
Saisir N

Initialisations :
Tracer le segment reliant les points de coordonnée (0 ; 0) et (1 ; 1)
Tracer le segment reliant les points de coordonnée (0 ; 0) et (1 ; 0)
Tracer le segment reliant les points de coordonnée (1 ; 0) et (1 ; 1)
H prend la valeur $\frac{1}{N}$
P prend la valeur 1

Traitement et sortie :
Pour *i* allant de 1 à N **Faire**
  Tracer le segment reliant les points de coordonnée (1 ; P) et (P ; P)
  P prend la valeur P – H
FinPour

```
: Eff Dessin
: Ligne( 0,0,1,1)
: Ligne(0,0,1,0)
: Ligne(1,0,1,1)
: Prompt N
: 1/N→H
: 1→P
: For(I,1,N)
: Ligne(1,P,P,P)
: P – H →P
: End
```

Algorithme 3

Entrée :
Saisir N

Initialisations :
Tracer le segment reliant les points de coordonnée (0 ; 0) et (1 ; 1)
Tracer le segment reliant les points de coordonnée (0 ; 0) et (1 ; 0)
Tracer le segment reliant les points de coordonnée (1 ; 0) et (1 ; 1)
H prend la valeur $\frac{1}{N}$
P prend la valeur 0
X prend la valeur 1

Traitement et sortie :
Pour *i* allant de 1 à N **Faire**
  Tracer le segment reliant les points de coordonnée (P ; 0) et (1 ; X)
  P prend la valeur P + H
  X prend la valeur X – H
FinPour

```
: Eff Dessin
: Ligne( 0,0,1,1)
: Ligne(0,0,1,0)
: Ligne(1,0,1,1)
: Prompt N
: 1/N→H
: 1→P
: For(I,1,N)
: Ligne(1,P,P,P)
: P – H →P
: End
```

Pour aller plus loin (algorithme d'Augustin Coudray)

Variables :
n, k : entiers naturels

Entrée :
Saisir n

Traitement :
Effacer dessin
Pour i variant de 1 à n **Faire**
  Tracer le segment reliant les points de coordonnées (k ; 0) et (0 ; $n + 1 – k$)
  Tracer le segment reliant les points de coordonnées (– k ; 0) et (0 ; $n + 1 – k$)
  Tracer le segment reliant les points de coordonnées (k ; 0) et (0 ; – ($n + 1 – k$))
  Tracer le segment reliant les points de coordonnées (– k ; 0) et (0 ; – ($n + 1 – k$))
FinPour