complexité
Examen

Calcul de Π

Dans un travail pratique de deug mass 2ème année, il est demandé de calculer la valeur de π avec l'algorithme d'*Archimède* (287-212 av JC) : on approche la circonférence du cercle de rayon 1 par des polygones réguliers inscrits de 2^n cotés :

Le calcul s'exprime par le système récurrent suivant :

$H_0 = 0$, $H_{n+1} = \sqrt{\dfrac{1+H_n}{2}}$ H_n est la distance du centre du cercle (apothème) aux cotés du polygone constitué de 2^n cotés.

$P_0 = 2$, $P_{n+1} = \dfrac{P_n}{H_{n+1}}$ P_n est le demi périmètre du polygone inscrit de 2^n cotés.

Les étudiant ont répondu de deux façons :

```
function pi1( n : integer) : real ;
var H, P : real ;
      i : integer ;
begin
      H :=0 ; P := 2 ;
      for i=1 to n do
      begin
            H := sqrt((1+H)/2) ;
            P := P/H ;
      end ;
      pi := P ;
end ;
```

```
function H( n : integer) : real ;
begin
      if n=0
      then H :=0
      else H := sqrt((1+H(n-1))/2)
end ;
function P( n : integer) : real ;
begin
      if n=0
      then P :=2
      else P := P(n-1)/H(n) ;
end ;
function pi2( n : integer) : real ;
begin
      pi2 := P(n) ;
end ;
```

Question 1.

On demande d'évaluer les coûts respectifs des calculs de π avec *pi1* et avec *pi2*.

Multiplication de matrices.

Question 2.

La méthode élémentaire de multiplication de matrices se programme de la façon suivante :

```
type matrice = array[1..Nmax] [1..Nmax] of int ;
```

```
procedure  produit( int n; m1 : matrice; m2 : matrice; var r : matrice) ;
var i, j, k : integer ;
begin
      for i := 1 to n do
      for j := 1 to n do
      begin
            r[i, j] :=0;
            for k :=1 to n do r[i, j] := r[i, j] + m1[i, k]*m2[k, j];
      end
end ;
```

Compter le nombre d'opérations arithmétiques effectuées pour multiplier deux matrices $n \times n$.

Question 3.

Nous faisons l'hypothèse que $n>1$ est une puissance de 2. L'algorithme de multiplication de matrices de *Strassen*(1969) est le suivant : soient a et b les matrices à multiplier, et r la matrice produit. Chaque matrice est divisée en 4 sous-matrices $n/2 \times n/2$.

$$a = \begin{bmatrix} a1 & a2 \\ a3 & a4 \end{bmatrix} \qquad b = \begin{bmatrix} b1 & b2 \\ b3 & b4 \end{bmatrix}$$

On calcule ensuite les 7 matrices $n/2 \times n/2$:

$$\left.\begin{aligned} m1 &= (a1+a4)\times(b1+b4) \\ m2 &= (a3+a4)\times b1 \\ m3 &= a1\times(b2-b4) \\ m4 &= a4\times(b3-b1) \\ m5 &= (a1+a2)\times b4 \\ m6 &= (a3-a1)\times(b1+b2) \\ m7 &= (a2-a4)\times(b3+b4) \end{aligned}\right\}$$

Les produits de matrices sont calculés avec l'algorithme de Strassen

On a alors :

$$r = \begin{bmatrix} m1+m4-m5+m7 & m3+m5 \\ m2+m4 & m1+m3-m2+m6 \end{bmatrix}$$

On demande de calculer le nombre d'opérations arithmétiques effectuées par cet algorithme.

Question 4.

On considère la suite $(a_n)n\geq 0$ définie par : $a_0 = a_1 = 0$ et $a_n=(n+1)n(n-1)$ pour $n\geq 2$

1) Calculer de deux manières différentes la fonction (série) génératrice associée à la suite $(a_n)n\geq 0$

- En considérant les fonctions (séries) génératrices associées aux suites (n^3) et (n).
- En développant en série génératrice la fonction $\dfrac{3!}{(1-x)^4}$

2) Calculer 3.2.1 + 4.3.2 + ... +(n+1)n(n-1)