Introduction à l’algorithmique
- première partie -

Vasile-Marian SCUTURICI

Procédures et fonctions

- Algorithme
  - élaboré par une démarche descendante
  - décomposer le problème en sous-problèmes, chaque sous-problème devant être de nouveau spécifié puis résolu

$\rightarrow$réutilisabilité

$\rightarrow$un algorithme peut être “appelé” dans le corps d’un autre algorithme

$\rightarrow$algorithme = procédure

Paramètres effectifs

- appel d'un algorithme :
  - préciser les valeurs des paramètres en entrée,
  - récupérer les valeurs des paramètres en sortie

$\rightarrow$ paramètres effectifs.

Exemple $\sum_{i=1}^{n} \cos(i)$

Procédure *somme-cos*(n, sc)

- **Entrée** : entier n
- **Sortie** : entier sc
- **Précondition** : $n \geq 1$
- **Postcondition** : $sc = \sum_{i=1}^{n} \cos(i)$
  ou, autrement dit, $sc = \cos(1) + \cos(2) + \cos(3) + \ldots + \cos(n)$

Procédure *cos*$(x, eps, cosx)$

- **Entrée** : réel x, eps
- **Sortie** : réel $cosx$
- **Précondition** : $eps > 0$
- **Postcondition** : $cosx = cos(x)$, avec une précision de eps
  (utilise le développement en série : $cos(x) = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i}}{(2i)!}$)

Exemple

Procédure *somme-cos*(n, sc)

...

Déclaration : réel $i, cosi$

const $eps = 0.0001$

début

$sc \leftarrow 0$

pour $(i \leftarrow 1\,;\, i \leq n\,;\, i \leftarrow i+1)$ **faire**

`// invariant:` $sc = \sum_{k=1}^{i-1} \cos(k)$

cos$(i, eps, cosi)$

$sc \leftarrow sc + cosi$

Complexité : ?

Exemple $\sum_{i=1}^{n} i!$

Procédure *factorielle*(n, f)
    Déclaration : entier i
    début
        $f \leftarrow 1$
        $i \leftarrow 1$
        tant que $i \leq n$ **faire**
            // invariant : $f = (i-1)!$
            $f \leftarrow f * i$
            $i \leftarrow i + 1$

Complexité : ?

Exemple $\sum_{i=1}^{n} i!$

Procédure *somme-fact*(n, sf)

- **Entrée** : entier n
- **Sortie** : entier sf
- **Précondition** : $n \geq 1$
- **Postcondition** : $sf = \sum_{i=1}^{n} i!$
  ou, autrement dit, $sf = 1! + 2! + 3! + ... + n!$
- **Déclaration** : entier i, fi

début

```
    sf ← 1
    i ← 1
    tant que i < n faire
        // invariant: sf = Σ_{k=1}^{i} k!
        i ← i + 1
        factorielle(i, fi)
        sf ← sf + fi
```

Complexité : ?

Exemple somme-fact

début

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

  $i \leftarrow i + 1$

  factorielle(i, fi)

  $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

  Déclaration : entier i

  début

    $f \leftarrow 1$

    $i \leftarrow 1$

    tant que $i \leq n$ **faire**

      $f \leftarrow f * i$

      $i \leftarrow i + 1$

somme(3,x)	
n	3
sf	?
i	?
fi	?

Exemple somme-fact

début

$\boxed{sf \leftarrow 1}$

$i \leftarrow 1$

tant que $i < n$ **faire**

$i \leftarrow i + 1$

factorielle(i, fi)

$sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

Déclaration : entier i

début

$f \leftarrow 1$

$i \leftarrow 1$

tant que $i \leq n$ **faire**

$f \leftarrow f * i$

$i \leftarrow i + 1$

somme(3,x)	
n	3
sf	1
i	?
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

somme(3,x)	
n	3
sf	1
i	1
fi	?

Exemple somme-fact

début

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

  $i \leftarrow i + 1$

  factorielle(i, fi)

  $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

  Déclaration : entier i

  début

    $f \leftarrow 1$

    $i \leftarrow 1$

    tant que $i \leq n$ **faire**

      $f \leftarrow f * i$

      $i \leftarrow i + 1$

somme(3,x)	
n	3
sf	1
i	1
fi	?

Exemple somme-fact

début
$sf \leftarrow 1$
$i \leftarrow 1$
tant que $i < n$ **faire**
$i \leftarrow i + 1$
factorielle(i, fi)
$sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
Déclaration : entier i
début
$f \leftarrow 1$
$i \leftarrow 1$
tant que $i \leq n$ **faire**
$f \leftarrow f * i$
$i \leftarrow i + 1$

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

  $i \leftarrow i + 1$

  factorielle(i, fi)

  $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

  Déclaration : entier i

  début

    $f \leftarrow 1$

    $i \leftarrow 1$

    tant que $i \leq n$ **faire**

      $f \leftarrow f * i$

      $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	?
i	?

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	1
i	?

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	1
i	1

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

$i \leftarrow i + 1$

factorielle(i, fi)

$sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

Déclaration : entier i

début

$f \leftarrow 1$

$i \leftarrow 1$

tant que $i \leq n$ **faire**

$f \leftarrow f * i$

$i \leftarrow i + 1$

fact(2,fi)	
n	2
f	1
i	1

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	1
i	1

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	1
i	2

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	1
i	2

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	2
i	2

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	2
i	3

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	2
i	3

somme(3,x)	
n	3
sf	1
i	2
fi	?

Exemple somme-fact

début

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

  $i \leftarrow i + 1$

  factorielle(i, fi)

  $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

  Déclaration : entier i

début

  $f \leftarrow 1$

  $i \leftarrow 1$

  tant que $i \leq n$ **faire**

    $f \leftarrow f * i$

    $i \leftarrow i + 1$

fact(2,fi)	
n	2
f	2
i	3

somme(3,x)	
n	3
sf	1
i	2
fi	**2**

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $\boxed{sf \leftarrow sf + fi}$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

somme(3,x)	
n	3
sf	3
i	2
fi	2

Exemple somme-fact

début

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

  $i \leftarrow i + 1$

  factorielle(i, fi)

  $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

Déclaration : entier i

début

  $f \leftarrow 1$

  $i \leftarrow 1$

  tant que $i \leq n$ **faire**

    $f \leftarrow f * i$

    $i \leftarrow i + 1$

somme(3,x)	
n	3
sf	3
i	2
fi	2

Exemple somme-fact

début

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

$\boxed{i \leftarrow i + 1}$

factorielle(i, fi)

$sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

Déclaration : entier i

début

$f \leftarrow 1$

$i \leftarrow 1$

tant que $i \leq n$ **faire**

$f \leftarrow f * i$

$i \leftarrow i + 1$

somme(3,x)	
n	3
sf	3
i	3
fi	2

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
  Déclaration : entier i
début
- $f \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i \leq n$ **faire**
  - $f \leftarrow f * i$
  - $i \leftarrow i + 1$

...

fact(3,fi)	
n	3
f	?
i	?

somme(3,x)	
n	3
sf	3
i	3
fi	2

Exemple somme-fact

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
- **Déclaration** : entier i
- **début**
  - $f \leftarrow 1$
  - $i \leftarrow 1$
  - **tant que** $i \leq n$ **faire**
    - $f \leftarrow f * i$
    - $i \leftarrow i + 1$

fact(3,fi)	
n	3
f	6
i	4

somme(3,x)	
n	3
sf	3
i	3
fi	2

Exemple somme-fact

début

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

  $i \leftarrow i + 1$

  factorielle(i, fi)

  $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

  Déclaration : entier i

début

  $f \leftarrow 1$

  $i \leftarrow 1$

  tant que $i \leq n$ **faire**

    $f \leftarrow f * i$

    $i \leftarrow i + 1$

fact(3,fi)	
n	3
f	6
i	4

somme(3,x)	
n	3
sf	3
i	3
fi	**6**

Exemple somme-fact

début
$sf \leftarrow 1$
$i \leftarrow 1$
tant que $i < n$ **faire**
$i \leftarrow i + 1$
factorielle(i, fi)
$sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)
Déclaration : entier i
début
$f \leftarrow 1$
$i \leftarrow 1$
tant que $i \leq n$ **faire**
$f \leftarrow f * i$
$i \leftarrow i + 1$

somme(3,x)	
n	3
sf	9
i	3
fi	6

Exemple somme-fact

début

- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - factorielle(i, fi)
  - $sf \leftarrow sf + fi$

somme-fact$(3, x)$

Procédure *factorielle*(n, f)

Déclaration : entier i

début

- $f \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i \leq n$ **faire**
  - $f \leftarrow f * i$
  - $i \leftarrow i + 1$

somme(3,x)	
n	3
sf	9
i	3
fi	6

?

$\sum_{i=1}^{n} i!$ – version plus performante

Procédure *somme-fact*(n, sf)

Entrée : entier n

Sortie : entier sf

Précondition : $n \geq 1$

Postcondition : $sf = \sum_{i=1}^{n} i!$

ou, autrement dit, $sf = 1! + 2! + 3! + ... + n!$

Déclaration : entier i, f

début

$f \leftarrow 1$

$sf \leftarrow 1$

$i \leftarrow 1$

tant que $i < n$ **faire**

// $f = i!$ et $sf = \sum_{j=1}^{i} j!$

$i \leftarrow i + 1$

$f \leftarrow f * i$

$sf \leftarrow sf + f$

Complexité : ?

Paramètres en entrée et en sortie

$$\boxed{n \leftarrow 4}$$
$$factorielle(n, n)$$

…	
n	4

Paramètres en entrée et en sortie

$n \leftarrow 4$

$\boxed{factorielle(n, n)}$

…	
n	24

Procédure *nom-algo*(var)

Entrée/Sortie : type-de-var var

var^{in}

var^{out}

Paramètres en entrée et en sortie

Procédure *échanger*(a, b)

Entrée/Sortie : entier a, b

Postcondition : $a^{out} = b^{in}$

$b^{out} = a^{in}$

Déclaration : entier aux

début

$aux \leftarrow a$

$a \leftarrow b$

$b \leftarrow aux$

Fonction

- Fonction : procédure dont le but est de calculer une seule valeur.

Fonction *type-fct nom-fct*$(<$ paramètres $>)$

Entrée : liste des paramètres (types et noms)

Précondition : Conditions sur les paramètres en entrée

Postcondition : relation entre la valeur retournée par la fonction et ses paramètres en entrée

- Dans le corps de la fonction, la sortie est spécifiée par :

retourner *expr*

Fonction - exemple

Fonction *entier factorielle*(n)

Entrée : entier n

Précondition : $n \geq 0$

Postcondition : retourne $n!$

Déclaration : entier i, f

début

$f \leftarrow 1$

pour $(i \leftarrow 1\,; i \leq n\,; i \leftarrow i+1)$ **faire**

// invariant: $f = (i-1)!$

$f \leftarrow f * i$

// Nombre de passages $= n\,;\ i = n+1$ et $f = (i-1)! = n!$

retourner f

$x \leftarrow$ factorielle(4)

$y \leftarrow$ factorielle$(x - 3 * \text{factorielle}(3))$

x = ? y = ?

Passage de paramètres

début
- $sf \leftarrow 1$
- $i \leftarrow 1$
- **tant que** $i < n$ **faire**
  - $i \leftarrow i + 1$
  - $fi \leftarrow i$
  - factorielle(fi)
  - $sf \leftarrow sf + fi$

- si *factorielle(n)*

Procédure *factorielle*(n)
- **Entrée/Sortie** : entier n
- **Déclaration** : entier i, f
- **début**
  - $f \leftarrow 1$
  - **pour** $(i \leftarrow 1 \,;\, i \leq n \,;\, i \leftarrow i + 1)$ **faire**
    - $f \leftarrow f * i$

somme-fact$(3, x)$

fact(3,fi)	
n	
f	?
i	?

somme(3,x)	
n	3
sf	3
i	3
fi	2

Récursivité

- Stratégie de résolution de problèmes algorithmiques où un problème est divisé en sous-problèmes de même type
- *Divide et impera* (diviser et conquérir)

Exemple - factorielle

$n! = 1 * 2 * \ldots * (n-1) * n$

$fact(n) = fact(n-1) * n$

Fonction $Fact(n)$
- **Entrée** : entier n
- **Sortie** : entier $factoriel$
- **Précondition** : $n \geq 0$
- **Postcondition** : $factoriel = n!$
- **début**
  - **si** $n = 0$ **alors**
    - $factoriel \leftarrow 1$
  - **sinon**
    - $factoriel \leftarrow n * Fact(n-1)$
  - **retourne** $factoriel$

Exemple - Fibonacci

Fonction *entier fibo*(n)

Entrée : entier n

Précondition : $n \geq 0$

Postcondition : retourne $fibo(n)$

où $fibo$ est la suite définie récursivement par :

$$fibo(0) = 1$$

$$fibo(1) = 1$$

$$fibo(n) = fibo(n-1) + fibo(n-2), \forall n \geq 2$$

début

si $n \leq 1$ **alors**

retourner *1*

sinon

retourner *fibo_rec(n-1)+fibo_rec(n-2)*

Complexité : ? **$\Theta(2^n)$**

Exemple – x^y

Fonction : entier puissance_rec(x, y)
Entrées :
  reel x
  entier y
Précondition :
  $y \geq 0$
Postcondition :
  retourne x^y
début
  si $y = 0$ **alors**
    retourner *1*
  sinon
    retourner $x *$ puissance_rec($x, y - 1$)
  finsi
fin

Complexité : ?

Exemple – x^y

$$x^{n+n} = x^n * x^n$$
$$x^{n+n+1} = x^n * x^n * x$$

Complexité : ?

$\Theta(\log_2(y))$

Fonction : entier puissance_rec_opt(x, y)
Entrées :
  reel x
  entier y
Déclarations :
  entier p
début
  si $y = 0$ **alors**
    retourner *1*
  sinon
    $p \leftarrow$puissance_rec_opt(x, y div 2)
    si y *mod* $2 = 0$ **alors**
      retourner $p * p$
    sinon
      retourner $p * p * x$
    finsi
  finsi
fin

Exemple – recherche dichotomique récursive

Fonction : entier cherche_dicho_rec(tab, d, f, e)
Entrées :

Telt[?.. ?] tab

entier d, f

Telt e

Précondition :

$\forall i \in [d..f-1],\ tab[i] \leq tab[i+1]$

Postcondition :

si $\exists i \in d..f$ tel que $tab[i] = e$ alors retourne i sinon retourne -1

Exemple – recherche dichotomique récursive

Fonction : entier cherche_dicho_rec(tab, d, f, e)
Déclarations :
  entier $milieu$
début
  si $d > f$ **alors**
    retourner *-1*
  sinon
    $milieu \leftarrow (d + f)/2$
    si $tab[milieu] = e$ **alors**
      retourner $(milieu)$
    sinon si $tab[milieu] < e$ **alors**
      retourner $cherche\_dicho(tab, milieu + 1, f, e)$
    sinon
      retourner $cherche\_dicho(tab, d, milieu - 1, , e)$
    finsi
  finsi
fin

Complexité : ?

$\Theta(\log_2(f\text{-}d+1))$

Récursivité

- Avantage
  - Elégance et compréhensibilité du code
- Désavantage
  - Chaque appel récursif peut nécessiter des ressources supplémentaires (mémoire + processeur)

To iterate is human, to recurse, divine.

L. Peter Deutsch

?

Exercice 1

- Proposez un algorithme pour multiplier deux matrices A et B
  - réel A[0..n-1, 0..m-1]
  - réel B[0..m-1, 0..p-1]

Exercice 2

- Proposer un algorithme pour multiplier n matrices