

Algorithmique TD1 (correction): généralités, invariants de boucle, coût d'un algorithme

15 octobre 2008

1. Ecrire un algorithme qui demande à l'utilisateur de saisir des nombres positifs jusqu'à ce qu'il rentre une valeur négative, puis qui retourne la valeur et le rang du plus grand nombre saisi.

```
programme saisie
variables
  val: entier
  max: entier
  rang: entier
  compteur: entier

debut
  ecrireln(" Entrer une valeur entière")
  lire (val)

  max ← val
  compteur ← 1
  rang ← 1

  tant que (val ≥ 0) faire
    ecrireln(" Entrer une valeur entière")
    lire (val)
    compteur ← compteur + 1
    si (val > max) alors
      max ← val
      rang ← compteur
    fsi
  ftq

  ecrireln("La plus grande valeur saisie est "&val&" en "&rang
    &" ieme position")
fin
```

2. Le grand classique : factorielle :
 - (a) Ecrire une fonction qui calcule la factorielle d'un entier passé en paramètre
 - (b) Trouver un invariant de boucle pour votre fonction factorielle et démontrer la validité du résultat.

(c) Evaluer le coût de votre fonction factorielle.

```

fonction factorielle(E val:entier):entier
variables
  res:entier
  compteur:entier

  res ← 1 //1 instruction
pour(compteur,2,val,1) //3 instructions (incrément 2, test
  1)
  {res = factorielle(compteur - 1)}
  res ← res * compteur //2 instruction
fpour //2 instruction
  {res = factorielle(compteur)}
  //(val - 1) boucles
  {res = factorielle(compteur) et
  compteur=val}
retourner res {res = factorielle(val)}
finfonction //cout: (val-1)*5 + 1

```

3. Un autre classique : pgcd

- (a) Ecrire une fonction qui calcul le pgcd de deux entiers passés en paramètres.
- (b) Trouver un invariant de boucle pour votre fonction pgcd et démontrer la validité du résultat.
- (c) Evaluer le coût de votre fonction pgcd.

```

fonction pgcd(E a:entier,E b:entier):entier
variables
  x:entier
  y:entier

  x ← a //1 instruction
  y ← b //1 instruction
  tantque (x ≠ y) faire //1 instruction
  {(pgcd(x,y) = pgcd(x-y,y) et (x>y))
  ou (pgcd(x,y)= pgcd(x,y-x) et (y>
  x))}
  si (x>y) alors {pgcd(x,y) = pgcd(x-y,y) et (x>y)}
  x ← x - y //2 instructions
  sinon {pgcd(x,y) = pgcd(x,y-x) et (y>x)}
  y ← y - x //2 instructions
  fsi
ftq //max(a,b) boucles
  {pgcd(x,y)=x et pgcd(a,b)=x}
retourner x //cout: max(a,b)*3+2
finfonction

```