IN 303

→ 4-4) Complexité

EXO 4 : 4) multiple(n : entier) : entier

entrée : n
sortie : somme des multiples de 3 et 5 jusque à n

début :

i ← 0
somme ← 0

tant que i ≤ n faire
si i modulo 3 = 0 alors
somme ← somme + i
sinon si i modulo 5 = 0 alors
somme ← somme + i
fin si
i = i + 1
fin tant que
fin
retourner somme

2) 4 opérations arithmétiques
→ nombre d'itérations : n
au total : 4n opérations arith.
→ O(n)

$$\sum_{i=4}^{n} 3i + \sum_{i=4}^{n} 5i - \sum_{i=4}^{n} 15i$$

$\rightarrow 3\frac{n(n+1)}{2} + 5\frac{n(n+1)}{2} - 15\frac{n(n+1)}{2}$

Exo 2 :

Algo 1 → 1 aff
à chaque itération : 1 aff
→ nombre d'itérations est $n/2$
au total : $1 + n/2 \times 1 = n/2 + 1 \rightarrow O(n)$

Algo 2 → init 2 aff
boucle → $(n/2) \times 2$
au total : $2 + 2(n/2) = 2 + n \rightarrow O(n)$

Algo 3 → nombre aff ($i \leftarrow i+1$)
nombre itérations $= n$
total : n aff

nombre aff - boucle (1)
1 itération : 2 aff -
n aff -
total : $2 + n$ aff
→ nombre itérations $= n$
total : $n(2+n)$ aff

Total : $1 + 2n + n^2$ aff -
$\rightarrow O(n^2)$

Algo 4 → $\sum_{k=1}^{n-1} = (n-1) + (n-2) + (n-3) + \dots + 3 + 2 + 1$

$\sum_{k=1}^{n-1} k = \frac{(n-1)n}{2}$ → nombre itérations : $= n$

$\rightarrow 2n + \frac{n(n-1)}{2}$ $\rightarrow O(n^2)$

Algo_5 → O(max(n, m)) + 1

Algo_6 → O(max(n, m)) + 1

Algo_7 → m + n

Algo_8 → O(n m)

Exo 4:

1) min (t: tableau)
début:
i ← 0
min ← T[0]
tant que i < t - 1 faire
si t[i] < min
min ← T[i]
fin si
i ← i + 1
fin tant que
retourner min

2) nombre de comparaisons: t - 1

3) 1ère façon: je modifie les valeurs du tableau avec (-t)
2ème façon: j'inverse (t[i] < min) → (t[i] > max)

4) On compare n valeurs entre elles puis on compare la + petite à la précédente et la + grande des deux avec la + grande actuelle. On a donc (3n)/2

6)
```
h(n:entier):entier
début:
        i ← 0 ; masc1 ← T[0];
        masc2 ← T[0];
4) tant que i < T.n faire
        si T.i > masc1
            masc2 ← masc1
            masc1 ← T.i
8)      fin si
        i = i+1
    fin tant que
    fin
    retourner masc1; masc2;
```

coût = [illegible]

Exo 3 : Algo 9 →

n	x	
1	1	$2^1 - 1$
2	3	$2^2 - 1$
3	7	$2^3 - 1$
4	15	$2^4 - 1$

→ $2^n - 1$

n correspond nombre de bits et x à la valeur max possible en base 10.

Algo 10 →

n	
1	1
2	3
3	7
4	15

Il calcul la même chose que l'Algo 9

$$\sum_{i=1}^{n} 2^i = 2^{n+1} - 1$$