

TD 1

calculs et complexité

Calculs exacts

Exercice 1 Il y a 20 ans un ordinateur faisait 10 millions d'opérations par seconde, et implementait un algorithme de tri qui demandait $50n \log_{10}(n)$ opérations. On le compare à un ordinateur 100 fois plus rapide et à un algorithme demandant n^2 opérations. Quels sont les temps de calcul pour chacun sur une entrée d'un million de nombres? et pour dix millions?

Exercice 2 Montrer par induction la formule :

$$\sum_{i=1}^n i^2 = \frac{1}{6}n(2n+1)(n+1)$$

Exercice 3 Montrer par induction la formule :

$$\sum_{k=0}^n k^3 = n^2(n+1)^2/4$$

Exercice 4 Calculez les sommes ou termes des suites suivantes :

1. $\sum_{i=1}^n \sum_{j=1}^i 1$
2. $\sum_{i=1}^n \sum_{j=i}^n 1$
3. $c_0 = 1$, et $c_{n+1} = c_n + 4$
4. $d_0 = 4$, et $d_{n+1} = d_n - 3$
5. $e_0 = 1$; $e_1 = 3$, et $e_{n+1} = 2e_n - e_{n-1} + 5$
6. $f_n = \sum_{i=1}^{n-1} (i - f_i)$, et $f_1 = 1$

Exercice 5 On positionne les entiers suivant un triangle :

```

1
2 3
4 5 6
7 8 9 10
...
```

Donner une expression précise de la somme des entiers d'une ligne i .

Exercice 6 Evaluer la fonction de complexité du bout de programme suivant :

```

Pour i de 1 à n-1 faire
  Pour j de i+1 à n faire
    Pour k de 1 à j faire
      {instruction en cout constant}
```

Exercice 7 Quelle est la valeur renvoyée par l'algorithme suivant, en fonction de n ? Quelle est la complexité en temps de l'algorithme? Et en mémoire?

```

Enigme (n : entier)
  x := 0
  pour i de 1 à n faire
    x := x + 2
    pour j de 1 à n faire
      pour k de 1 à j faire
        x := x + 1
      fin pour
    fin pour
  fin pour
  renvoyer x

```

Exercice 8 Donner en fonction de n la complexité en temps dans le pire des cas de la procédure suivante :

```

Procédure Oddi donc (n : entier)
var i, j, x, y : entier;
begin
  Pour i de 1 à n faire
    Si i est impair alors
      begin
        Pour j de 1 à n faire x <- x+1;
        Pour j de 1 à i faire y <- y+1;
      end
    end
  end
end

```

Exercice 9 Donner la complexité de l'algorithme suivant en détaillant de manière exacte le nombre d'opérations élémentaires, puis en concluant :

```

FUNCTION amediter (Integer array: t)
BEGIN
  int x = 0;
  FOR 1 <= i <= longueur(t)
    FOR i <= j <= longueur(t) - i
      IF t[j] < t[i] THEN
        x = x + 1
      ELSE
        x = x - 1
      END IF
    ENDFOR
  ENDFOR
END

```