

# Complexité des algorithmes [cx]

## Exercices de cours

Karine Zampieri, Stéphane Rivière, Béatrice Amerein-Soltner

Unisciel  algoprogram  UNIVERSITÉ HAUTE-ALSACE Version 21 mai 2018

### Table des matières

<b>1</b>	<b>Appréhender le cours</b>	<b>2</b>
<b>2</b>	<b>Appliquer le cours</b>	<b>2</b>
2.1	Recherche du maximum . . . . .	2
2.2	Schéma de Hörner . . . . .	3
2.3	Tours de Hanoï . . . . .	4

### C++ - Exercices de cours (TP)



Mots-Clés Complexité des algorithmes ■

Difficulté ●●○ (1 h) ■

# 1 Appréhender le cours

## 2 Appliquer le cours

### 2.1 Recherche du maximum



#### Objectif

Cet exercice évalue la complexité en nombre de comparaisons de la recherche du maximum dans un `Tableau` défini comme suit :



#### Définitions C++

```
const int TMAX = ...;  
using ITableau = int[TMAX];
```



Écrivez une fonction `maximumTab(t,n)` qui calcule et renvoie la plus grande valeur des `n` premières valeurs d'un `ITableau t`.



Quelle est la complexité de votre algorithme en nombre de comparaisons ?



Montrez qu'il est optimal.

## 2.2 Schéma de Hörner



Utilise Complexités en temps ■

Durée estimée 15 min ■



### Schéma de Hörner

Le schéma de HÖRNER évalue la valeur d'un polynôme pour une valeur de la variable. Il est basé sur la réécriture :

$$\begin{aligned} P(x) &= a_0 + a_1x + \dots + a_nx^n \\ &= a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + x(a_n)) \dots)) \end{aligned}$$



Écrivez une fonction `evalHorner(p,n,x)` qui calcule et renvoie la valeur d'un polynôme `p` (tableau de réels) d'ordre `n` (entier) en `x` (réel).



Quelle est la complexité de votre algorithme en nombre d'opérations ?

## 2.3 Tours de Hanoï



Utilise Complexités en temps ■

Durée estimée 20 min ■



### Objectif

L'algorithme des TOURS DE HANOI est défini par :

Pour déplacer  $n$  disques de la tige A vers la tige C :

1. On déplace les  $(n - 1)$  plus petits de la tige A vers la tige B.
2. On déplace le plus gros disque de la tige A vers la tige C.
3. On déplace les  $(n - 1)$  plus petits de la tige B vers la tige C.

L'algorithme s'écrit comme suit :

```

Action hanoi ( n : Entier ; orig , dest , inter : Chaîne )
Début
| Si ( n > 0 ) Alors
|   | hanoi ( n - 1 , orig , inter , dest )
|   | déplacer ( orig , dest )
|   | hanoi ( n - 1 , inter , dest , orig )
| FinSi
Fin

```



On note  $H(n)$  le nombre de passages d'un disque d'un piquet à un autre. Donnez la relation de récurrence.



Résolvez la récurrence.



Déduisez la complexité de la procédure.



Concluez.