Travaux Dirigés sur le langage C n$^{\text{o}}$.3

Cours d'Informatique de Deuxième Année

—Deug SSM M3—

Nous abordons dans ce troisième *TD* la notion de récursivité. Les objectifs sont les suivants :

- Traduction en C d'un algorithme récursif,
- Introduction au calcul de complexité en temps.

La récursivité...

▸ **Exercice 1**. Donner une fonction `fact` qui calcule la factorielle d'un entier passé en argument. On rappelle la définition de la fonction $factorielle$:

$$\begin{cases} 0! & = & 1, \\ n! & = & n \times (n-1)!, \qquad n > 0. \end{cases}$$

On donnera aussi la complexité en temps de cette fonction.

▸ **Exercice 2**. On veut dans cet exercice calculer a^n au moyen d'une fonction. On dispose de trois formules qui réalisent ce calcul.

- ◇ $a^0 = 1$ et si $n > 0$, $a^n = a \times a^{n-1}$,
- ◇ $a^0 = 1$, et si $n > 0$, si n est pair alors $a^n = \left(a^2\right)^{\frac{n}{2}}$ sinon —n est impair— $a^n = a \times \left(a^2\right)^{\frac{n-1}{2}}$,
- ◇ $a^0 = 1$, et si $n > 0$, si n est pair alors $a^n = \left(a^{\frac{n}{2}}\right)^2$ sinon —n est impair— $a^n = a \times \left(a^{\frac{n-1}{2}}\right)^2$.

Programmer les trois versions. Apprécier la différence entre les deux dernières versions au niveau de l'appel récursif. Estimer la complexité en temps des trois fonctions.

▸ **Exercice 3**. Donner une fonction qui calcule le **pgcd** de deux entiers passés en argument. On utilisera le schéma récursif suivant :

$$\text{pgcd}(n,m) = \begin{cases} n & \text{si } m = 0, \\ \text{pgcd}(m, n\%m) & \text{sinon.} \end{cases}$$

pour lequel on supposera que m et n sont des entiers positifs non tous les deux nuls. On admettra que la complexité en temps est en $O\big(\log(\max(n,m))\big)$. Peut-on obtenir simplement un algorithme de calcul du **ppcm** de m et n ?

Tableau et récursivité...

▸ **Exercice 4**. On désire calculer simultanément le minimum et le maximum dans un tableau d'entiers,

1. en utilisant la méthode "naïve". Montrer que l'on fait au plus $2 \times (n-1)$ comparaisons entre les éléments du tableau,
2. en utilisant la méthode "dichotomique" —récursive—. Montrer que l'on fait au plus $\lceil \frac{3 \times n}{2} \rceil - 2$ comparaisons[1].

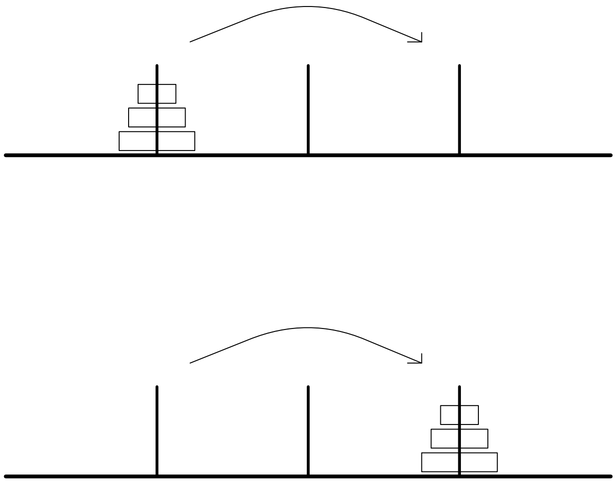
---

[1] C'est à dire $\lfloor \frac{n}{2} \rfloor$ comparaisons en moins.

Les tours de Hanoï...

▸ **Exercice 5**. On cherche à résoudre le problème des tours de Hanoï. Le problème consiste à déplacer n disques en suivant certaines contraintes, d'une tour à une autre en utilisant une troisième tour.

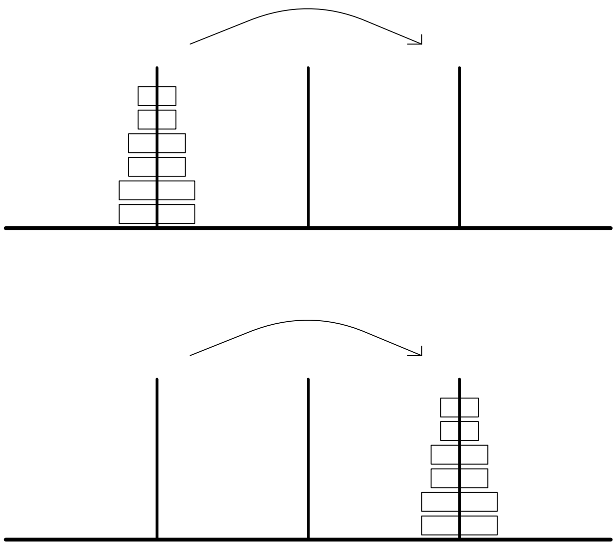
- Faire "tourner à la main" l'algorithme pour $n = 2$ et $n = 3$,
- Donner l'algorithme récursif de résolution du problème,
- En déduire la fonction correspondante, dont on donnera la complexité en temps.



Exemple du problème pour $n = 3$.

Exercice supplémentaire

▸ **Exercice 6**. On considère le problème des tours de Hanoï pour lequel chaque disque de taille t est présent p fois. On a par ailleurs n tailles différentes de disque. On demande dans un premier temps de résoudre le problème pour $p = 2$ puis d'en déduire l'algorithme général —et le programme C— correspondant à un nombre p quelconque d'occurences de chaque disque[2].



Exemple du problème pour $p = 2$ et $n = 3$.

[2] On a donc $n \times p$ disques...