Algorithmique
TD1. Semaine du 08/10/01

Exercice 1

Le but de cet exercice est de vous exercer à la manipulation des logarithmes et exponentielles. On rappelle que, par définition:

$\log_{10}(x) = \ln(x)/\ln(10)$

et plus généralement,

$\log_m(x) = \ln(x)/\ln(m)$ pour tout entier $m > 0$.

Rappelons également quelques propriétés des logarithmes

- le domaine de ln est $\mathcal{R}_+^*$,
- $\ln(x \times y) = \ln(x) + \ln(y)$ pour tous réels x, y,
- ln est strictement croissante,
- ln est une bijection de $\mathcal{R}_+^*$ dans $\mathcal{R}$.
- $p^{\log_p(x)} = x$ pour tout $x \in \mathcal{R}_+^*$.

A partir de ces propriétés,

1. Démontrer que pour tout entier $p \geq 1$, on a $\log_{10}(10^p) = p$.
2. Qu'est-ce qui est le plus grand, 10^{100} ou 100^{10} ?
3. Si $\log_{10}(n) = 5$, alors que vaut n ?
4. Déduire de la question 1 la valeur exacte du nombre de chiffres d'un entier n, en écriture décimale.

Exercice 2

- Trouver une solution exacte (forme close) pour la somme

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

  et prouver qu'elle est correcte..

- Prouver par récurrence que

$$\sum_{i=1}^{n} i^2 = 1 + 4 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

Exercice 3

On considère un tableau de taille n qui contient des entiers. Le tri par sélection consiste à regarder chaque emplacement du tableau l'un après l'autre, en recherchant à chaque fois quel est l'élément, parmi ceux qui ne sont pas encore triés, qui devrait s'y trouver pour qu'il le soit. Ici, on parcourt le tableau de gauche à droite et on veut trier les éléments en ordre croissant.

- Écrire l'algorithme du tri par sélection.
- Analyser l'algorithme dans le meilleur des cas:
  - calculer le nombre d'affectations effectués.
  - calculer le nombre de comparaisons effectués.
- Analyser l'algorithme dans le pire des cas:
  - calculer le nombre d'affectations effectués.
  - calculer le nombre de comparaisons effectués.

Exercice 4

On considère un tableau de taille n d'entiers 0 ou 1, représentant un entier en binaire, le bit de poids k étant écrit à la case k.

- Écrire une procédure en C réalisant l'algorithme habituel d'incrémentation.
- Calculer le coût pour les entiers compris entre 0 et 7.
- Donner, en fonction de l'entier représenté, le coût de cet algorithme dans le pire des cas.