Chapitre 1 : Qu'est ce qu'un algorithme ?

(sous-entendu algorithme déterministe)

Vocabulaire (proposition vague)

Un algorithme est une opération (ou une suite d'opérations) **non ambigüe** opérant sur une ou plusieurs structures formelles.

Exemple de structure formelle : mot binaire, tableau de mots, graphe, arborescence, ...

But d'un algorithme (à priori) : résoudre un problème formel.
Exemples de problème :

- Trouver le plus grand diviseur commun à deux nombres.
- Trouver l'écriture décimale de la somme de deux nombres (écrit en décimal).
- Trouver l'écriture romaine (ex MMXXII) de la somme de deux nombres (écrit en romain).

Exemple :

Voici deux "formes" d'un même algorithme : trouver le pgcd (plus grand diviseur commun) de deux nombres.

En langage C.

```
int euc_pgcd(int a,int b) {
    int c;
    while(b!=0) {
        c=a%b;
        a=b;
        b=c;
    }
    return a;
}
```

En langage moins technique

algorithme : euc_pgcd

entrées : a : entier, b : entier ; sortie : a : entier.

1. Si b est nul, on stoppe.
2. Si b n'est pas nul,
   - $(a, b) \leftarrow (b, r)$, r étant le reste de la division euclidienne de a par b
   - on saute à 1.

Remarques :

- Non ambigüité sur chaque étape à suivre.
- un algorithme possède éventuellement des entrées et éventuellement des sorties.
- Deux "exécutions" d'un algorithme avec de mêmes entrées doivent se dérouler exactement de la même manière.
  - Un algorithme n'est pas un protocole, pas une interface.
  - Les algorithmes parallélisables et algorithmes non déterministes sont des cas à part.
- Certaines descriptions d'algorithmes sont laxistes sur ce que font certaines opérations (pour des raisons de clarté ou autres). Exemple : somme de deux nombres. (nombre binaire, en chiffre romain)

Ambigüité sur les opérations

On rencontre des descriptions d'algorithmes incomplètes. Qui font à appelle à d'autre algorithme (ou plus souvent à une solution à un problème formel "facile") Ça remet en question la notion de non-ambigüité d'un algorithme.

- Certains bénin : (faire une somme, un produit)
- D'autres un peu moins :
  Pour trouver le maximum d'un tableau : trier le tableau par ordre croissant puis renvoyer le dernier élément.

Une raison principale : la description est allégée.
Un inconvénient : certains problèmes sont plus compliqués qu'en apparence.