

Chapitre 2 : Terminaison, Correction et Complexité d'un algorithme

Rappels :

- ▶ Un algorithme (déterministe)
 1. description (le plus possible) non-ambigüe.
 2. la valeur entrée détermine complètement le déroulement.

Analyse d'un algorithme

- ▶ Résout-il le/un problème ?
 - ▶ **Terminaison de l'algorithme** : Pour n'importe quelle entrée, l'algorithme se termine-t-il ? (Réponses possibles : Oui Non)
 - ▶ **Correction** : Pour n'importe quelle entrée, les sorties de l'algorithme sont-elles conformes au problème à résoudre ? (Réponses possibles : Oui Non)

Le problème : proposition logique sur les entrées et sorties de l'algorithme.

- ▶ Est-il performant ?
 - ▶ **Complexité en temps**. Combien d'opérations élémentaires l'algorithme effectue-t-il ?
 - ▶ **Complexité en espace** Quelle place en mémoire occupe-t-il ?

La réponse à ces deux questions est donnée sous la forme d'une suite numérique, où plus souvent on donne la classe de domination à laquelle appartient cette suite : $\Theta(\dots)$ ou $O(\dots)$.

Complexité

- ▶ **Dans le pire des cas** : on cherche un objet sur lequel l'algorithme effectuera le plus grand nombre d'opérations (ou la plus grande place occupée) parmi tous les objets de taille n .
- ▶ **En moyenne** : une moyenne du nombre d'opérations (ou une moyenne de la place occupée) selon une distribution de probabilité sur les objets de taille n . Il s'agit souvent d'une distribution uniforme.