Notions fondamentales

1 Exemple de domaines informatiques.

L'informatique ne se résume pas à l'utilisation d'ordinateurs. On peut citer Alan Turing, qui disait que "l'informatique n'est pas plus la science des ordinateurs que l'astronomie n'est la science des téléscopes". C'est une discipline très vaste, avec beaucoup de domaines, dont notamment :

- Les bases de données
- Les réseaux
- Les systèmes embarqués
- La robotique
- La sécurité des systèmes d'information
- La cryptographie
- Le calcul scientifique
- L'intelligence artificielle
- L'interface utilisateur
- Le développement web
- La recherche opérationnelle et l'optimisation
- La vérification de programmes
- La logique

2 Problèmes

Un **problème** est la donnée d'un format de **paramètres d'entrée** (qui décrit quelles données on a entrées), et de la **sortie attendue** en fonction de ces paramètres.

Exemple:

Rectangle

Entrée:	Un entier $a \in \mathbb{N}$, un entier $b \in \mathbb{N}$.
Sortie:	La surface d'un rectangle de longueur a et de largeur b.

Parité

Entrée:	$n \in \mathbb{N}$.
Sortie:	`Vrai` si n est pair, `Faux` sinon.

On appelle **problème de décision** un problème dans lequel la sortie attendue est `Vrai` ou `Faux`, et **problème d'optimisation** un problème dans lequel la sortie attendue minimise ou maximise une fonction donnée sur un ensemble donné.

Exemple: |**Parité** est un problème de décision. Les deux problèmes qui suivent sont des problèmes d'optimisation :

Plus court chemin

Entrée:	Un graphe G (pondéré), un point de départ $A \in G$ et un point d'arrivée $B \in G$.
Sortie:	Un chemin de longueur minimale de A à B dans G

Sac à dos

Entrée:	$n \in \mathbb{N}$, $P_{max} \in \mathbb{R}^+$, $(p_i)_{i\in[\![1,n]\!]} \in (\mathbb{R}^+)^n$, $(v_i)_{i\in[\![1,n]\!]}$
Sortie:	$(x_i^*)_{i\in[\![1,n]\!]} \in \{0,1\}^n$ tel que $\sum_{i=1}^n x_i^* p_i \leq P_{max}$ et maximisant f $= \begin{pmatrix} \{0,1\}^n & \to & \mathbb{R} \\ (x_1,...,x_n) & \mapsto & \sum_{i=1}^n x_i v_i \end{pmatrix}$

Pour un problème donné, une **instance** est une famille de paramètres répondant au type des entrées du problème.

Exemples :

- 18 est une instance positive (de solution `Vrai`) de **Parité**.
- 17 est une instance négative (de solution `Faux`) de **Parité**.
- (7,8) est une instance de **Rectangle** de solution 56.
- (1,(1,2),(3,15)) est un instance de **Sac à dos** de solution (1,0).

3 Algorithmes

Un **algorithme** est une méthode systématique à base d'**opérations** qui permet de résoudre toutes les instances d'un problème.

Exemple: Comptage binaire sur les doigts :

```
Commencer les doigts levés.
Si le pouce est baissé
    Lever le pouce
Sinon
    Baisser les doigts levés consécutifs au pouce
    Lever le doigt suivant
    Baisser le pouce
```

En programmation, une **fonction** peut être vue come un rassemblement d'instructions, qui s'appliquent sur des données que l'on peut faire varier avec ce que l'on appelle argument. Si on attend généralement d'une fonction qu'elle donne/retourne un résultat en sortie, mais elle peut aussi avoir des **effets de bord** : affichages, modification de la mémoire...

Remarque: Quitte à décomposer les problèmes difficiles en sous-problèmes, on essaiera d'avoir un fonction par problème. Pour que ce découpage par fonctions soit lisible dans le code, on munit chaque fonction dans celui-ci d'une spécification qui contient :

- La signature de la fonction, contenant :
  - Le nombre et le type des arguments
  - Le type de la sortie
- Les hypothèses sur les arguments
- La valeur de la sortie pour les arguments et éventuellement la description des effets de bord.

Exemple:

```c
int surface_rectangle(int longueur, int largeur)
{
    // hyp: longueur >= 0 && largeur >= 0
    /* Renvoie la surface d'un rectangle de longueur `longueur` et de largeur `largeur` */
    return longueur * largeur
}
```

4 Langage de programmation

Un langage de programmation est une manière conventionnelle et formelle de formuler des algorithmes

intelligibles par les humains et exécutables par une machine.

	Langue naturelle	Langage de programmation
Alphabet	`abc...zABC...Z 0123456789 .;:"()`	`+@{}#&...` (table ASCII)
Mots	Mots présents dans le dictionnaire, noms propres	Mots clés, identificateurs
Grammaire	Former des mots/phrases selon les règles de la grammaire	Syntaxe
Sémantique	Sens de la phrase	Sémantique