VI. Stratégie de résolution des problèmes

1. ***Approche par force brute:***
   - Résoudre directement le problème, à partir de sa définition ou par une recherche exhaustive
2. ***Diviser pour régner:***
   - Diviser le problème à résoudre en un certain nombre de sous-problèmes (plus petits)
   - Régner sur les sous-problèmes en les résolvant récursivement :
     - Si un sous-problème est élémentaire (indécomposable), le résoudre directement
     - Sinon, le diviser à son tour en sous-problèmes
     - Combiner les solutions des sous-problèmes pour construire une solution du problème initial

VI. Stratégie de résolution des problèmes

3. ***Programmation dynamique:***
   - Obtenir la solution optimale à un problème en combinant des solutions optimales à des sous-problèmes similaires plus petits et se chevauchant
4. ***Algorithmes gloutons:***
   - Construire la solution incrémentalement, en optimisant de manière aveugle un critère local

VII. Les classes de problèmes

Type de problèmes

Un problème est étroitement lié à la question posée dans sa définition. Selon la forme de la question posée, il existe différents types de problèmes selon le degré de leur difficulté :

- Problème de décision : Réponse 'oui' ou 'non'
- Problème de recherche de solution : Trouver une ou plusieurs solution possible
- Problème d'optimisation : calcul de la solution approchée
- Problèmes de dénombrement de solution : Déterminer le nombre de solutions du problème sans toutefois les rechercher

Les classes de problèmes

Définition 1:

Un algorithme est dit en temps polynômial, si pour tout n, n étant la taille des données, l'algorithme s'exécute en temps borné par un polynôme $f(n) = c*n^k$ opérations élémentaires.

Définition 2:

On dit qu'un problème est polynômial si est seulement si il existe un algorithme polynomiale pour le résoudre.

Définition 3:

Un algorithme est dit ***de résolution*** s'il permet de construire la solution au problème, et il est dit ***de validation*** s'il permet de vérifier si une solution donnée répond au problème.

Les classes de problèmes

Définition 4: Réduction polynomiale:

Un problème P1 peut être ramené ou réduit à un problème P2 si une instance quelconque de P1 peut être traduite comme une instance de P2 et la solution de P2 sera aussi solution de P1. la fonction de traduction ou de transformation doit être polynomiale (De complexité polynomiale).

La relation de réductibilité est réflexive et transitive

La réduction polynomiale de P1 en P2 est noté :

$$P1 \leq P2$$

VII. Les classes de problèmes

Les classes de problèmes

Quelle est la particularité d'un algorithme polynômial par rapport à un algorithme exponentiel ?

VII. Les classes de problèmes

La classe P

Un problème est de classe P s'il existe un algorithme ***polynomiale déterministe*** pour le résoudre.

La classe NP

Un algorithme est de classe NP si et seulement si, il existe un algorithme de validation ***non déterministe*** et qui s'exécute en un temps ***polynomiale***. Ainsi, la classe NP contient l'ensemble des problèmes dont la vérification est polynomial mais dont la résolution ne l'est pas obligatoirement.

Les problèmes NP-Complet

On dit qu'un problème A est NP-Complet si et seulement si:

1. A appartient à la classe NP
2. $\forall B \in NP, B \leq A$.

VII. Les classes de problèmes

Les problèmes NP-Complet

On dit qu'un problème A est NP-Complet si et seulement si:

1. A appartient à la classe NP
2. $\exists B \in NP - Complet, B \leq A$.

Explication : B est NP-Complet $\Rightarrow \forall X \in NP, X \leq B$

Donc si on arrive à trouver un problème déjà démontrer NP-Complet et qu'on puisse le réduire en notre problème A, on aura démontrer par transitivité que tous les problèmes NP peuvent être réduit en le problème A. Donc, si :

$$\exists B \in NP - Complet, B \leq A \Rightarrow \forall X \in NP, X \leq B \leq A$$

VII. Les classes de problèmes

- Quelques problèmes NP-complets
  - SAT : le père des problèmes NP-complets (le tout premier à avoir été montré NP-complet par Stephen COOK en 1971)
  - 3-SAT : Satisfiabilité d'une conjonction de clauses dont chaque clause est composée d'exactement trois littéraux
  - CYCLE HAMILTONIEN : Existence dans un graphe d'un cycle hamiltonien
  - VOYAGEUR DE COMMERCE : Existence dans un graphe pondéré d'un cycle hamiltonien de coût minimal
  - CLIQUE : Existence dans un graphe d'une clique (sous-graphe complet) de taille k
  - 3-COLORIAGE D'UN GRAPHE : peut-on colorier les sommets d'un graphe avec trois couleurs de telle sorte que deux sommets adjacents n'aient pas la même couleur ?
  - PARTITION : Peut-on partitionner un ensemble d'entiers en deux sous-ensembles de même somme ?