Centre Universitaire de Mila

Institut des sciences et de la technologie

1ère Master STIC

Module : Optimisation combinatoire

Série de TD1 : théorie de complexité

**Exercice 1 : Complexité algorithmique**

1. Ecrire l’algorithme permettant de vérifier si un nombre entier est premier ou non. Evaluer sa complexité.
2. Ecrire l’algorithme permettant de calculer la somme d’une matrice de n\*n entiers. Evaluer sa complexité.
3. Quelle est la différence principale entre ces deux problèmes.
4. Donner les classes de complexité des deux algorithmes précédents.

**Exercice 2 : Classes de complexité**

1. Citer les différentes classes de complexités.
2. Généralement on dit que le temps d’exécution d’un algorithme est polynomial ou exponentiel, classer les classes précédent selon ces deux classes.

**Exercice 3 : Classes de problèmes**

1. Expliquer par un exemple la notion de la classe NP
2. Expliquer par un exemple la notion de la classe NP-Complet
3. Expliquer par un exemple la notion de la classe NP-Difficile

**Exercice 4 : Algorithme non déterministe**

Soit E l’ensemble de n éléments tel que $E=\{a\_{1}, a\_{2}, …a\_{n}\}$

Ecrire un algorithme non déterministe (NP) de complexité O(n) qui détermine s’il existe ou pas un sous ensemble s, S ⊂ E telque $\sum\_{i=1}^{n}a\_{i}=M$

**Exercice 5 : Problèmes de décision**

1. Donner le problème de décision associé pour chacun des deux problèmes suivant :
	1. Le problème de sac à dos 0-1
	2. Le problème de voyageur du commerce.
2. Donner un algorithme non déterministe pour l’un de ces deux problèmes.