**Exercices TD (Feuille 1)**

**Consignes :** Les exercices mis en évidence sont obligatoires pour valider le chapitre 1. Les autres exercices sont optionnels. Il vous est conseillé de les faire après les exercices obligatoires.

**1.1.1** *\_* Donnez un exemple concret qui intègre l’un des problèmes suivants : tri, optimisation de l’ordre de multiplication des matrices, détermination de l’enveloppe convexe.

**1.1.2 *\_***À part la vitesse, qu’est-ce qui pourrait servir à mesurer l’efficacité dans un contexte concret ?

**1.1.3** *\_* Sélectionnez une structure de données que vous avez déjà vue, puis étudiez ses avantages et ses inconvénients.

**1.1.4** *\_* En quoi le problème du chemin minimal et celui du voyageur de commerce, précédemment mentionnés, se ressemblent-ils ? En quoi sont-ils différents ?

**1.1.5** *\_* Trouvez un problème concret pour lequel seule conviendra la solution optimale. Trouvez ensuite un problème pour lequel une solution « approchée » pourra faire l’affaire.

**1.2.1** Donnez un exemple d’application exigeant des algorithmes intrinsèques, puis discutez les fonctions des algorithmes concernés.

**1.2.2** On veut comparer les implémentations du tri par insertion et du tri par fusion sur la même machine. Pour un nombre *n* d’éléments à trier, le tri par insertion demande 8*n*2 étapes alors que le tri par fusion en demande 64*n* lg *n*. Quelles sont les valeurs de *n* pour lesquelles le tri par insertion l’emporte sur le tri par fusion ?

**1.2.3** Quelle est la valeur minimale de *n* pour laquelle un algorithme dont le temps d’exécution est 100*n*2 s’exécute plus vite qu’un algorithme dont le temps d’exécution est 2*n* sur la même machine ?

**1.1. Comparaison de temps d’exécution**

Pour chaque fonction *f* (*n*) et pour chaque durée *t* du tableau suivant, déterminez la taille maximale *n* d’un problème susceptible d’être résolu dans le temps *t*, en supposant que l’algorithme mette *f* (*n*) microsecondes pour traiter le problème.

