# Licence Info
# Algorithmique Avancée
# TD 1: 26-27/9/05

1. Utiliser l'algorithme décrit en cours (celui de Karatsuba) pour mutiplier les deux entiers `172 204 370 251` et `284 160 293 042`; on considère ces nombres comme des nombres de 4 chiffres à base 1000; les multiplications de base de deux nombres inférieurs à 1000 se feront à la main ou par calculatrice.

2. L'algorithme *minmax* utilise $3n/2-2$ comparaisons pour trouver le minimum et maximum de $n$ nombres ($n$ pair).

   (a) Combien sont utilisées quand $n$ est impair?

   (b) Pour l'algorithme plus naif qui compare chaque nombre après le deuxième avec le maximum des précédents et (s'il était inférieur au maximum) avec le minimum, le nombre de comparaisons dépend de l'ordre des données; Quels sont les nombres minimum et maximum de comparaisons?

   (c) On veut démontrer que tout algorithme basé sur les comparaisons doit utiliser au moins le même nombre que *minmax*. On va utiliser une méthode de **l'adversaire** : chaque fois que l'algorithme effectue une comparaison, l'adversaire donne une réponse qui est cohérente avec les réponses précédentes mais choisie afin d'obliger l'algorithme à prendre le maximum de temps; ici l'adversaire devrait essayer de garder le plus grand nombre possible d'éléments du tableau qui pourraient être le minimum et/ou le maximum.
   Donner une stratégie pour l'adversaire qui oblige tout algorithme à effectuer au moins le nombre de comparaisons de *minmax*.
   Conclure que tout algorithme nécessite ce nombre, au moins dans le pire cas.

   (d) Pour étudier le comportement d'un algorithme en moyenne, on peut supposer que tous les $n!$ ordres possibles des $n$ données ont la même probabilité; avec cette supposition, est-il vrai que tout algorithme doit faire autant de comparaisons que *minmax* en moyenne?

3. On considère la taille maximum de la pile utilisée par le tri rapide.

   (a) Démontrer que pour tout ordre des $n$ données, cette taille est forcement au moins $log_2 n$

   (b) Avec la même supposition sur les probabilités de données que dans l'exercice précédent, et en utilisant le fait que la taille est au moins $1+$ la taille de la pile utilisée par l'appel sur la plus grande des deux parties, démontrer un minorant de l'espérance de la forme $k\ log_2 n$ ($k > 1$).

4. Sur une page web j'ai trouvé la remarque suivante sur l'algorithme de Strassen (je traduis de l'anglais mais les auteurs ne sont pas des anglophones) *Aujourd'hui ce résultat n'a aucune importance réelle pour les matrices de réels ou nombres complexes parce que nos ordinateurs effectuent une multiplication dans presque le même temps qu'une addition. Donc, il ne fait pas de sens de remplacer 8 multiplications et 4 additions par 7 multiplications et 18 additions.*

   (a) Expliquez pourquoi cette remarque est complètement stupide!

   (b) Un algorithme raisonnable utiliserait la méthode de Strassen pour les grandes matrices et basculerait sur la méthode simple pour les petites; si on ne compte que le temps des opérations de base (addition multiplication et soustraction des éléments des matrices) et que toutes ces opérations prennent le même temps, quel est le bon seuil entre les grandes et les petites matrices?