

TD 02 –  $\div \Rightarrow \text{crown}$ **Exercice 1.**

Justin Bieber

Dans un groupe de  $n$  personnes (numérotées de 1 à  $n$  pour les distinguer), une *star* est quelqu'un qui ne connaît personne mais que tous les autres connaissent. Pour démasquer une star, s'il en existe une, vous avez juste le droit de poser des questions à n'importe quel individu  $i$  du groupe, du type « est-ce que vous connaissez  $j$  ? ». On suppose que les individus répondent toujours la vérité. On veut un algorithme qui trouve une star s'il en existe une, et qui garantit qu'il n'y en a pas sinon, en posant le moins de questions possibles.

1. Combien peut-il y avoir de stars dans le groupe ?
2. Écrire le meilleur algorithme que vous pouvez et donner sa complexité en nombre de questions (on peut y arriver en  $\mathcal{O}(n)$  questions).
3. Donner une borne inférieure sur la complexité (en nombre de questions) de tout algorithme résolvant le problème.  
(Difficile) Prouver que la complexité exacte de ce problème est  $3n - \lfloor \log_2(n) \rfloor - 3$ .

**Exercice 2.**

S'il te plait

Une *matrice de Tœplitz* est une matrice  $n \times n$  ( $a_{i,j}$ ) telle que  $a_{i,j} = a_{i-1,j-1}$  pour  $2 \leq i, j \leq n$ .

1. La somme de deux matrices de Tœplitz est-elle une matrice de Tœplitz ? Et le produit ?
2. Trouver un moyen d'additionner deux matrices de Tœplitz en  $\mathcal{O}(n)$ .
3. Comment calculer le produit d'une matrice de Tœplitz  $n \times n$  par un vecteur de longueur  $n$  ? Quelle est la complexité de l'algorithme ?

**Exercice 3.**

Mehdi-Anne

Soit  $A$  un ensemble de  $n$  éléments distincts totalement ordonné. On cherche à trouver la médiane de l'ensemble, c'est-à-dire l'élément de rang  $\lfloor \frac{n+1}{2} \rfloor$ .

1. Donner un algorithme naïf. Quelle est sa complexité ?  
Pour améliorer la complexité de l'algorithme, on utilise la stratégie suivante : on regroupe les éléments par paquets de 5, on calcule la médiane de chaque paquet, puis la médiane des médianes.
2. Comment terminer cet algorithme ? Quelle est sa complexité ?
3. Que se passe-t-il si on regroupe par paquets de 3 ? Et de 7 ?

**Exercice 4.**<http://www.droite-rencontre.com/>

Un site internet cherche à regrouper ses membres en fonction des goûts musicaux de chacun. Pour cela, chaque membre doit classer par ordre de préférence une liste d'artistes<sup>1</sup>. On dit que deux membres, Arthur et Béatrice, ont des goûts musicaux proches lorsque qu'il y a peu d'*inversions* dans leurs classements : une inversion est une paire d'artiste  $\{L, M\}$  telle qu'Arthur préfère  $L$  à  $M$  et Béatrice préfère  $M$  à  $L$ . On cherche donc à compter le nombre d'inversion dans les classements d'Arthur et Béatrice.

1. Compter le nombre d'inversion les classements suivants :  
**Arthur** : Britney Spears, Lady Gaga, Michael Jackson, Madonna, Céline Dion ;  
**Béatrice** : Lady Gaga, Madonna, Britney Spears, Michael Jackson, Céline Dion.
2. Proposer un algorithme naïf qui résout le problème. Quelle est sa complexité ?

1. Le classement est un ordre total.

On cherche maintenant à améliorer l'algorithme précédent en utilisant le paradigme Diviser-Pour-Régner. Pour cela, on coupe le classement de chaque membre en deux sous-classements de même taille, celui des artistes préférés (classement supérieur) et celui des autres artistes (classement inférieur). On compte alors les inversions  $(L, M)$  qui peuvent être de deux types : soit  $L$  et  $M$  apparaissent dans le même sous-classement de Béatrice, soit  $L$  et  $M$  apparaissent dans deux différents sous-classements de Béatrice (inversions mixtes).

3. On suppose que les deux sous-classements de Béatrice sont triés en fonction du classement d'Arthur<sup>2</sup>. Montrer qu'on peut alors compter les inversions mixtes en temps linéaire.
4. Donner un algorithme de type Diviser-Pour-Régner qui fonctionne en temps  $\mathcal{O}(n \log n)$ .

---

2. Si  $L$  et  $M$  apparaissent dans le même sous-classement de Béatrice, alors ils apparaissent dans le même ordre que dans le classement d'Arthur.