

Centre Universitaire de Mila
Institut des sciences et de la technologie
Département de mathématiques et informatique

Master 1 I2A

Année : 2022/2023

Module : Algorithmique avancée et complexité

TD 4

Exercice 1 : algorithmes probabilistes

Etant donné un ensemble E de n entiers tel que : $i \in E$ et $j \in E$ $i \neq j$. Soit E^k un ensemble de combinaisons tel que chaque combinaison $c = (e_1, e_2, \dots, e_k)$ est composée de k éléments de E . On dit que la combinaison $c = (e_1, e_2, \dots, e_k) \in E^k$ est une combinaison ordonnée si : $e_1 \leq e_2 \leq \dots \leq e_{k-1} \leq e_k$. Soit O^k un sous-ensemble de E^k qui contient toutes les combinaisons ordonnées. On considère que les entiers de E sont stockés dans une liste L et nous avons déjà des fonctions suivantes :

- Ajouter (L, i) : qui ajoute un entier e dans la liste L ;
- Supprimer (L, i) : qui supprime un entier e de la liste L ;
- Aléatoire (L) : qui retourne un entier sélectionné aléatoirement à partir de la liste L ;

En utilisant ces fonctions, écrire un algorithme probabiliste qui estime la valeur du ratio r où :

$$r = |O^k|/|E^k| ?$$

Exercice 2 : Diviser pour régner

Soit P un ensemble de points dans un espace de deux dimensions (pour chaque $i \in P$, nous avons x_i et y_i). Notre objectif est de trouver les deux points les plus proches dans P . L'algorithme naïf qui teste toutes les possibilités pour trouver ces deux points a une complexité en $O(n^2)$.

Proposer un algorithme de type **diviser pour régner** qui trouve les deux points les plus proches dans P ? (La complexité de votre algorithme doit être meilleure que la complexité de l'algorithme naïf)