

Centre Universitaire de Mila
Institut de mathématiques et informatique
Département de l'informatique

Master 1 I2A

Année : 2024/2025

Module : Algorithmique avancée et complexité

TD 4

Exercice 1 : algorithmes probabilistes

Etant donné un tableau T de n entiers tel que : $\forall i, j : T[i] \neq T[j]$. Soit E^k l'ensemble de combinaisons générées à partir de T tel que chaque combinaison $c = (e_1, e_2, \dots, e_k)$ est composée de k éléments de T ($e_1 \neq e_2 \neq e_3, \dots, \neq e_k$). On dit que la combinaison $c_d \in E^k$ est une combinaison dominée par un entier β ssi : $e_1 + e_2 + \dots + e_{k-1} + e_k < \beta$. Soit D_β^k l'ensemble des combinaisons dominée par β dans E^k ($D_\beta^k \subset E^k$) et $\Delta_\beta = |D_\beta^k| / |E^k|$.

- 1) Ecrire un algorithme (une fonction) probabiliste qui permet d'estimer la valeur de Δ_β pour un paramètre β donné.
- 2) Quel est le type de votre algorithme ?
- 3) Donner l'ordre de grandeur de la complexité de votre algorithme ?
- 4) Il est clair que la valeur de Δ_β est liée à la valeur de β . Ecrire un algorithme probabiliste qui cherche la valeur de β qui rend Δ_β le plus proche possible de 0.5 (pour une valeur de k donnée par l'utilisateur)

Exercice 2 (facultatif) : 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)