TD 1

Rappel.

Inégalité de Markov Si X est une variable aléatoire ≥ 0, $\mathbb{P}[X \geq c\mathbb{E}[X]] \leq 1/c$ pour tout $c > 0$.

Borne de Chernoff Si $X_1, \ldots, X_n$ sont des variables aléatoires indépendantes à valeur dans $\{0,1\}$, et $X = \sum_{i=1}^n X_i$, alors $\mathbb{P}[X \leq (1-\delta)\mathbb{E}[X]] \leq e^{-\frac{\delta^2\mathbb{E}[X]}{2}}$ pour tout $\delta > 0$.

World's Most Useful Inequality Pour tout x, $(1+x) \leq e^x$.

Exercice 1. *`RandMinCut` amélioré*

Le but de cet exercice est d'étudier une variante de l'algorithme `RandMinCut` qui a une bien meilleure complexité. L'idée est que les problèmes arrivent plutôt quand il reste peu de sommets : le risque de contracter deux sommets *par erreur* devient plus grand.

1. On fixe une coupe minimum d'un graphe G à n sommets, et on contracte aléatoirement des arêtes de G jusqu'à ce que G ait $(n/\sqrt{2}+1)$ sommets. Montrer qu'avec probabilité au moins $1/2$, aucune arête de la coupe minimum n'a été contractée.

On note `Contraction`(G) l'algorithme qui contracte aléatoirement des arêtes de G jusqu'à ce que G ait $(n/\sqrt{2}+1)$ sommets. L'algorithme `BetterMinCut` est le suivant :

(a) Si G a moins de 8 sommets, renvoyer une coupe minimale ;
(b) $C_1 \leftarrow$ `BetterMinCut(Contraction(`G`))`;
(c) $C_2 \leftarrow$ `BetterMinCut(Contraction(`G`))`;
(d) Renvoyer $\min(C_1, C_2)$.

On suppose qu'on a un algorithme calculant une coupe minimale exacte pour un graphe de moins de 8 sommets (par exemple par recherche exhaustive).

2. Estimer la complexité de `BetterMinCut`.
3. En notant p_n la probabilité que `BetterMinCut` renvoie une coupe minimale quand le graphe en entrée a n sommets, borner p_n en fonction de $p_{n/\sqrt{2}+1}$.
4. *On admet que $p_n \geq \Omega(1/\log n)$.* Montrer qu'avec $O(\log^2 n)$ répétition de l'algorithme, sa probabilité de succès est au moins $1 - 1/n^c$ pour une certaine constante c. Quelle est la complexité globale obtenue ?

Exercice 2. *Meilleure analyse de `Quicksort`*

On dit qu'une partition d'un tableau T en deux sous-tableaux disjoints est *équilibrée* si aucun des deux sous-tableaux ne contient plus de $3/4$ des éléments de T.

1. Considérons un tableau T de taille n, et z un élément de ce tableau. Considérons la suite des partitions effectuées lors de l'exécution de `Quicksort` aux sous-tableaux dont z fait partie. Combien de partitions équilibrées suffisent pour z soit dans un sous-tableau de taille 1 ?
2. Soit Z_i la variable aléatoire qui vaut 1 si à l'étape i de récursion, le sous-tableau dont z fait partie est partitioné de manière équilibrée. Calculer $\mathbb{E}[Z_i]$.
3. Calculer $\mu = \mathbb{E}[\sum_{i=1}^{c\log n} Z_i]$ où $c \geq 0$ est une constante quelconque.
4. En déduire que si $c/4(1-2/c\log(4/3))^2 \geq 2$, la probabilité que z soit seul dans son sous-tableau après $c\log n$ étapes de récursion est supérieure à $1 - 1/n^2$. *Utiliser la borne de Chernoff.*
5. En déduire que `Quicksort` effectue $O(n\log n)$ étapes de récursions avec probabilité $\geq 1 - 1/n$.