TD 1

Exercice 1. *De Monte Carlo à Las Vegas*

Soit Π un problème pour lequel on dispose des deux algorithmes suivants : un algorithme A de type Monte Carlo, de complexité $T(n)$ et dont la probabilité de succès est (au moins) $p(n)$ (pour une entrée de taille n) ; un algorithme (déterministe) B de complexité $V(n)$ pour vérifier une solution pour Π.

✎ Construire un algorithme de type Las Vegas pour Π à partir des algorithmes A et B, et analyser l'espérance de son temps de calcul.

Exercice 2. QuickSelect *particulier*

1. Analyser l'espérance du nombre de comparaisons effectuées par QuickSelect$(T, 1)$ où T est un tableau de n éléments. *Indication. On supposera que l'espérance est bornée par $an + b$ pour des constantes a et b, et on déduira des valeurs correctes pour a et b.*
2. Même question pour $k = 2$.
3. Donner des algorithmes déterministes (simples !) pour ces deux cas, et comparer les complexités obtenues.

Exercice 3. 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_0 sommets. Quelle est la probabilité qu'une arête de la coupe minimale ait été contractée ?
2. Montrer que si $n_0 \geq 1 + n/\sqrt{2}$, 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 tant que G a plus de $(1 + n/\sqrt{2})$ sommets. L'algorithme BetterMinCut est le suivant.

BetterMinCut(G) :
1. Si G a moins de 8 sommets, renvoyer une coupe minimale ;
2. $C_1 \leftarrow$ BetterMinCut(Contraction(G)) ;
3. $C_2 \leftarrow$ BetterMinCut(Contraction(G)) ;
4. 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).

3. Montrer que la complexité de BetterMinCut est $O(n^2 \log n)$.
4. 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 $p_{n/\sqrt{2}+1}$.
5. *On admet que $p_n = \Omega(1/\log n)$.* Montrer qu'avec $O(\log^2 n)$ répétitions 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 ?