Modèles de calcul, Complexité, Approximation et Heuristiques

Important : Durée : 1h.
- *Tous les exercices sont indépendants.*
- *Vos réponses doivent être courtes mais clairement argumentées ou commentées.*
- *Tous documents manuscrits et polycopiés autorisés.*

1 Algorithmes et complexité probabilistes

Cette partie est composée de 3 exercices indépendants.

1. Complexité probabiliste. (2 points)
1. Montrer que $ZPP = RP \cap co - RP$.
2. Montrer que $P \subset RP \subset NP$.

2. Isomorphisme d'arbres (3 points) Deux arbres $T_1 = (V_1, E_1)$ et $T_2 = (V_2, E_2)$ sont dits isomorphes si et seulement si il existe une bijection $f : V_1 \to V_2$ telle que : pour tout nœud $v \in V_1$ ayant pour fils $w_1, \ldots, w_k$ dans T_1, alors les fils de $f(v) \in V_2$ sont exactement $f(w_1), \ldots, f(w_k)$ dans T_2.
1. La hauteur $h(v)$ d'un nœud dans un arbre est définie par :
   - si v est une feuille, $h(v) = 0$;
   - sinon soient $w_1, \ldots, w_k$ les fils de v, alors : $h(v) = 1 + \max\{h(w_1), \ldots, h(w_k)\}$.

   Vérifier brièvement que deux arbres isomorphes ont exactement le même nombre de fils de chaque hauteur.
2. Soit H la hauteur de la racine de T_1. Soient les $H + 1$ indéterminées $X_0, \ldots, X_H$, une associée à chaque hauteur dans l'arbre. A chaque nœud v de T_1, on associe un polynôme $P(v)$ défini par :
   - Si v est une feuille : $P(v) = X_0$;
   - sinon soient $w_1, \ldots, w_k$ les fils de v, alors :
   $$P(v) = \left(X_{h(v)} - P(w_1)\right)\left(X_{h(v)} - P(w_2)\right)\cdots\left(X_{h(v)} - P(w_k)\right).$$
   Soit P_1 (respectivement P_2) le polynôme associé à la racine de T_1 (respectivement T_2). On admet que T_1 et T_2 sont isomorphes si et seulement si $P_1 = P_2$.
   Soit a l'arité maximale de T (tout nœud de T_1 a au plus a fils). Vérifier que $\deg(P_1) \leq a^H$.
3. Donner un algorithme probabiliste qui teste si deux arbres sont isomorphes ; votre algorithme est-il Atlantic City, Monte Carlo ou Las Vegas ? Préciser son nombre d'opérations et sa probabilité d'erreur.

3. Sélection d'un élément (5 points) Soit $X = [x_1, \ldots, x_n]$ un tableau de n éléments supposés distincts d'un ensemble totalement ordonné. Pour un entier $k \in \{1, \ldots, n\}$, le problème SELECTION($X[1..n]$, k) consiste à trouver l'élément de rang k dans X ; c'est à dire calculer la valeur $\bar{x} \in X$ telle que exactement $k - 1$ éléments de X sont strictement inférieurs à $\bar{x}$.
Par exemple, SELECTION($X[1..n]$, $n/2$) retourne l'élément médian de X.
On considère l'algorithme Monte Carlo suivant pour SELECTION($X[1..n]$, k) :
- Choisir aléatoirement et uniformément $n^{3/4}$ éléments de X qui forment un sous-ensemble Y.
- Trier Y : dans la suite on note $y_1 < y_2 < \ldots < y_{n^{3/4}}$ les éléments de Y triés.
- Soit $p := \max\{1, \lfloor kn^{-1/4} - \sqrt{n}\rfloor\}$ et soit $a := y_p$;
  soit $g := \min\{\lceil kn^{-1/4} + \sqrt{n}\rceil, n^{3/4}\}$ et soit $b := y_g$.
- Comparer chacun des n éléments du tableau X à a et b pour déterminer le nombre n_a (respectivement n_b) d'éléments de X strictement inférieurs à a (respectivement b) et le tableau Z défini par :
  - cas $k < n^{1/4}$: si $k > n_b$ retourner ERREUR, sinon $Z := \{x \in X \text{ tels que } x \leq b\}$ et $n_a := 1$;
  - cas $k > n - n^{1/4}$: si $k < n_a$ retourner ERREUR sinon $Z := \{x \in X \text{ tels que } x \geq a\}$;
  - cas $k \in \{n^{1/4}, \ldots, n - n^{1/4}\}$: si $k < n_a$ ou $k > n_b$ retourner ERREUR ; sinon $Z := \{x \in X \text{ tels que } a \leq x \leq b\}$.
- Soit $|Z|$ le nombre d'éléments de Z ; si $|Z| > 4n^{3/4} + 2$, retourner ERREUR ;
- Trier Z ; on note $z_1 < z_2 < \ldots < z_{|Z|}$ les éléments de Z triés.
- Retourner z_{k-n_a+1}.

Questions :
1. Vérifier que le coût de cet algorithme en nombre de comparaisons est $2n + o(n)$.
2. Que retourne l'algorithme si $k < n^{1/4}$ ou si $k > n - n^{1/4}$ et quel est son coût ?
   NB : dans toute la suite, on suppose $k \in \{n^{1/4}, \ldots, n - n^{1/4}\}$.
3. Justifier que si $\bar{x} \notin Z$, alors $\bar{x} < a$ ou $b < \bar{x}$. Montrer brièvement que la probabilité de cet évènement est $O\left(n^{-1/4}\right)$.
4. On suppose ici que $\bar{x} \in Z$ et on pose $k_a = \max\{1, k - 2n^{3/4}\}$ et $k_b = \min\{k + 2n^{3/4}, n\}$. Justifier que si $|Z| > 4n^{3/4} + 2$, alors nécessairement le rang de l'élément a dans X est inférieur à $k - 2n^{3/4}$ ou le rang de l'élément b dans X est supérieur à $k + 2n^{3/4}$.
   Montrer brièvement que la probabilité de cet évènement est $O\left(n^{-1/4}\right)$.
5. Transformer l'algorithme précédent en algorithme Las Vegas : majorer le coût de cet algorithme en nombre de comparaisons et majorer sa probabilité d'échec.
6. Transformer cet algorithme en un algorithme qui donne toujours le bon résultat ; majorer le nombre moyen de comparaisons qu'il effectue quelle que soit l'instance en entrée..
7. Donner un algorithme déterministe de temps moyen $O(n)$ (dérandomisation).
8. Modifier l'algorithme de l'énoncé pour qu'il effectue $n + \min\{k, n-k\} + o(n)$ comparaisons. Cette optimisation a-t-elle une influence sur les variantes Las Vegas et dérandomisée proposées dans les questions précédentes ?