N° etudiant : **Nom :** **Prénom :**

Partiel - Sujet A

Consignes : Les réponses doivent tenir sur la feuille (il y a largement assez d'espace). Aucun document n'est autorisé. Toute tentative de fraude sera sanctionnée.

Afin de ne pas perdre de points bêtement, lisez bien le sujet en entier avant de commencer. Écrivez le code au crayon à papier, les copies trop sales seront pénalisées. Il y a 5 exercices, le barême est indicatif et peut être modifié.

Exercice 1 (2 points)**.** Étant donné une fonction récursive dont la complexité peut s'exprimer récursivement sous la forme $T(n) = 8T(n/8) + \mathcal{O}(n)$, quelle est sa complexité ? Vous justifierez soit à l'aide d'une preuve, soit à l'aide d'un théorème vu en cours.

Exercice 2 (3 points)**.** Écrivez les relations entre les fonctions suivantes en termes de $\mathcal{O}()$ et de $\Theta()$:

$$f(n) = 2n^2 \log^{1200} n + 0,01n^4 + 1000n$$
$$g(n) = 20n^3 + n^4$$
$$h(n) = n^{2n}$$

N° etudiant : **Nom :** **Prénom :**

Exercice 3 (5 points). Soit $f : [0,1] \to \mathbb{R}$ une fonction continue croissante telle que $f(0) = -2$ et $f(1) = 2$, on cherche à approximer x, la solution de l'équation $f(x) = 0$. Écrivez un algorithme qui étant donné n en entrée détermine une approximation y à 2^{-n} près de x, c'est à dire que $|y - x| \leq 2^{-n}$. La complexité de cet algorithme devra être en $\mathcal{O}(n)$ appels à f.

Exercice 4 (5 points)**.** Soit $G = (V, E)$ un graphe orienté, écrivez un algorithme qui étant donné un ensemble $S \subseteq V$ et un sommet $s \in V$ détermine la distance de l'ensemble S au sommet s. Donnez la complexité de votre algorithme.

Exercice 5 (5 points)**.** Un arbre binaire de recherche (ABR) est un arbre dans lequel pour chaque noeud de valeur v, les valeurs de son sous-arbre droit sont supérieures à v et les valeurs du sous-arbre gauche sont inférieures à v. Un arbre est équilibré quand son sous-arbre gauche et son sous-arbre droit ont la même taille à un noeud ou une feuille près. Écrire un algorithme qui étant donné un tableau trié T renvoie un ABR équilibré contenant les valeurs de v. Vous pourrez supposer que vous avez accès à deux fonctions :

- `Arbre(v,A1,A2)` qui construit un noeud de valeur v avec comme sous-arbre gauche `A1` et comme sous-arbre droit `A2`. `A1` et `A2` doivent être soit des arbres, soit des feuilles, soit `Vide`.
- `Feuille(v)` qui construit une feuille de valeur v.
- `Vide` qui construit un arbre vide.

Vous donnerez la complexité de votre algorithme en fonction de la taille du tableau T.