Christophe Mouilleron
christophe.mouilleron@ensiie.fr

# Devoir à la maison – Racines entières et factorisation

**Consignes** :
- Vous pouvez travailler en binôme.
- Déposez votre code et votre rapport (au format PDF) dans le dépôt ae_dm avant le jeudi 30 mai 2019 à 23h59.
- N'oubliez pas d'inclure les graphiques produits avec Gnuplot dans votre rapport, et de fournir les fichiers de données qui vous ont permis de faire ces graphiques.

## 1 Calcul de la racine carrée entière

Dans un premier temps, on va utiliser la méthode de Newton pour calculer la racine carrée entière d'un entier naturel $a$.

**Question 1.** Donnez l'itération de Newton correspondant à la fonction $f : x \mapsto x^2 - a$.

**Question 2.** En déduire un algorithme permettant de calculer une valeur approchée de $\sqrt{a}$.
**note :** On pourra, dans un premier temps, utiliser $x_0 = a$ comme valeur initiale.

**Question 3.** Pour éviter les calculs numériques, on va plutôt utiliser l'itération

$$r \leftarrow \left\lfloor \left\lfloor r + \frac{a}{r} \right\rfloor / 2 \right\rfloor .$$

On s'arrête alors dès que la nouvelle valeur obtenue n'est pas strictement plus petite que l'ancienne valeur.
Implantez cet algorithme à l'aide de GMP, et vérifiez sur les valeurs $a = 10, 48, 49, 50, 100$ et $1000$ que le résultat obtenu est $\lfloor \sqrt{a} \rfloor$.

**Question 4.** Montrez que votre algorithme calcule effectivement $\lfloor \sqrt{a} \rfloor$. Pour cela, on commencera par montrer que la valeur calculée $r$ vérifie :
(i) $r^2 \le a$ (expliquez pourquoi $r \le \lfloor \lfloor r + \frac{a}{r} \rfloor / 2 \rfloor$, puis utilisez $\lfloor x \rfloor \le x$),
(ii) $r > \sqrt{a} - 1$ (montrez d'une part que $g : x \mapsto \left(x + \frac{a}{x}\right) / 2$ admet un minimum global, et d'autre part que $\lfloor \lfloor x \rfloor / 2 \rfloor > x/2 - 1$ pour tout réel $x$).

**Question 5.** Étudiez le coût de votre implantation. On tracera la courbe du temps d'exécution en fonction du nombre de limbs de $a$ avec Gnuplot, et on commentera le résultat obtenu dans le rapport.

**Question 6.** Avez-vous des idées pour améliorer les performances de votre code ? Si oui, expliquez-les dans votre rapport. Vous êtes aussi invités à les implanter et comparer le temps d'exécution par rapport à celui du code initial.

## 2 Généralisation à la racine $k$-ième

On va maintenant calculer la racine $k$-ième entière d'un grand entier positif $a$, c'est à dire $\lfloor \sqrt[k]{a} \rfloor$. Notez qu'il s'agit du plus grand entier $r$ tel que $r^k \le a$.

**Question 7.** On va utiliser cette fois la fonction $f_k : x \mapsto x^k - a$. Donnez l'itération de Newton correspondant à la fonction $f_k$.

**Question 8.** Expliquez comment adapter l'algorithme vu à l'exercice précédent pour calculer des racines $k$-ième entières, où $k$ est maintenant une entrée de l'algorithme.

**Question 9.** Implantez à l'aide de `GMP` une nouvelle fonction permettant de calculer des racines $k$-ième. Testez votre nouveau code.

**Question 10.** Prouvez la correction de votre algorithme, en vous inspirant de la preuve faite précédemment pour le cas $k = 2$.

**Question 11.** Étudiez le coût de votre implantation. On fixera une valeur $a$ très grande et on fera varier $k$, afin de tracer la courbe du temps d'exécution en fonction de $k$ dans `Gnuplot`. Commentez le résultat obtenu dans votre rapport.

## 3 Factorisation des puissances parfaites

On suppose maintenant que le nombre $a$ en entrée est une puissance parfaite. L'objectif est alors de trouver $r$ et $k$ tels que $a = r^k$, et où $k$ est le plus grand possible.

**Question 12.** Proposez, à partir de ce qui a été fait précédemment, un algorithme pour résoudre ce problème.
**note :** On a toujours $a = a^1$, donc le problème admet toujours au moins une solution.

**Question 13.** Implantez votre algorithme et testez-le. Pensez à joindre des exemples dans votre code et/ou dans votre rapport.

## 4 Un algorithme de factorisation d'entiers

On a vu dans la section précédente comment factoriser un entier $a$ lorsqu'il est de la forme $r^k$ avec $k > 1$. Si un nombre n'est ni premier, ni de la forme $r^k$, il s'écrit comme un produit de deux entiers $p$ et $q$ strictement plus grands que 1 et différents.

**Question 14.** On peut toujours se ramener au cas où $p$ et $q$ sont impairs, quitte à factoriser d'abord $a$ par la plus grande puissante de 2 possible.

Vérifiez que, dans ce cas, $a = \left(\dfrac{p+q}{2}\right)^2 - \left(\dfrac{p-q}{2}\right)^2$.

**Question 15.** Lorsque $p \approx q$, on a $\dfrac{p+q}{2} \approx \sqrt{a}$ et l'entier $\dfrac{p-q}{2}$ est une petite constante. On peut alors factoriser $a$ en procédant de la façon suivante :

1. On part de $u = \lceil \sqrt{a} \rceil$.
2. On calcule $u^2 - a$.
3. Si le résultat s'écrit sous la forme $v^2$, alors $p = u + v$, $q = u - v$ et $a = (u + v)(u - v)$.
4. Sinon, on incrémente $u$ et on repart au point 2.

Implantez cet algorithme et testez le sur quelques exemples pertinents.

**Question 16.** Après avoir rappelé la factorisation de $u^3 - v^3$, proposez un algorithme similaire à celui de la question précédente pour factoriser les entiers $a$ qui s'écrivent sous la forme d'un cube moins un petit cube. Implantez-le et illustrez son fonctionnement sur des exemples bien choisis.