TD10 : Tests de primalité

Exercice 1. *Premiers aléatoires*

1. Pour tester si un nombre N est premier, on tire aléatoirement un entier $k \leq \sqrt{N}$ et on teste si $\mathrm{pgcd}(k, N) \neq 1$.
   - **i.** Si N est un multiple de 3, quelle est la probabilité que $\mathrm{pgcd}(k, N) \neq 1$?
   - **ii.** Même question si $N = pq$ où p et q sont deux nombres premiers environ égaux à $\sqrt{N}$.
   - **iii.** Supposons que N est composé : quelle est (dans les deux cas) l'espérance du nombre de tirages nécessaires avant d'obtenir k tel que $\mathrm{pgcd}(k, N) \neq 1$?
2. Pour tirer un nombre premier aléatoire entre 2 et N, on tire un nombre aléatoire entre 2 et N et on teste s'il est premier. À l'aide du théorème des nombres premiers, donner l'espérance du nombre de tirage nécessaires pour trouver un nombre premier.

Exercice 2. *Test de puissances parfaites*

On souhaite tester si un entier N est une puissance parfaite, c'est-à-dire s'il s'écrit $N = n^e$ pour deux entiers n et $e \geq 2$.

1. Décrire un algorithme qui, étant donné N et e, détermine s'il existe un entier n tel que $n^e = N$. *Indication : chercher n par dichotomie.*
2. Montrer que si $N = n^e$, alors $e \leq \log_2(N)$.
3. En déduire un algorithme polynomial pour tester si N est une puissance parfaite et analyser sa complexité.

Exercice 3. *Exponentiation rapide*

On souhaite calculer $a^k \bmod N$, où a, k et N sont des entiers.

1. Écrire un algorithme naïf qui effectue $O(k)$ opérations sur les entiers. Comment s'assurer que la taille des entiers manipulés ne croisse pas trop rapidement ?
2. On souhaite un algorithme plus rapide, basé sur la technique « diviser-pour-régner ».
   - **i.** Exprimer $a^k \bmod N$ en fonction de $a^{\lfloor k/2 \rfloor} \bmod N$.
   - **ii.** En déduire un algorithme récursif qui calcule $a^k \bmod N$. *Faire attention à ce que les entiers manipulés ne croissent pas trop vite !*
   - **iii.** Calculer le nombre d'opérations sur les entiers effectués par votre algorithme.