TD7 : Tests de primalité

Exercice 1. *Produire des nombres premiers*

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, montrer que la méthode proposée fournit un algorithme Las Vegas pour produire un nombre premier, et donner l'espérance du nombre de tirage nécessaires.

Exercice 2. *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.

Exercice 3. *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 $n^e = N$.
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.