# Algorithmes probabilistes

F. Kany. ISEN-Brest & La Croix-Rouge

## Présentation

Un algorithme probabiliste est un algorithme utilisant des valeurs produites par un générateur de nombres pseudo-aléatoires.

On distingue :

1. les algorithmes de type Monte Carlo :
   On utilise la fonction `random` pour générer un très grand nombre $N$ de répliques indépendantes d'un phénomène. D'après von Mises : comme les probabilités représentent la limite de la fréquence d'observation d'un événement quand $N$ tend vers l'infini, on peut calculer une valeur approchée des probabilités avec un certain intervalle de confiance (cet intervalle diminue quand le nombre $N$ de répliques augmente). On obtient donc des probabilités approchées avec une complexité en temps certaine.
   Exemple : pour calculer le nombre $\pi$, on tire au hasard $N$ couples $(x, y)$ représentant les coordonnées de points dans un carré de coté 1 ; on cherche le nombre $P$ de points qui se trouve dans le quart de disque de rayon 1 ; on calcule $\alpha = P/N$. Si $N$ vaut quelques millions, $\alpha$ tend vers la probabilité que le point se trouve dans quart de disque. On obtient une assez bonne approximation de $\pi/4$ (surface du quart de disque/surface du carré) avec un temps de calcul certain (qui est proportionnel à $N$).
2. les algorithmes de type Las Vegas :
   On utilise la fonction `random` pour une étape interne d'un algorithme. On obtient un résultat exact (le même que l'algorithme non randomisé) mais avec une complexité qui est très probablement bien meilleure que l'algorithme non probabiliste.
   Exemple : le tri rapide (quicksort) est un algorithme qui trie une liste de $n$ nombres avec une complexité qui peut varier entre $O(n.\ln(n))$ et $O(n^2)$ (dans le pire des cas, si la liste est déjà triée). En randomisant la recherche du pivot, on obtient un tri exact avec une complexité qui, de façon quasi-certaine, est $O(n.\ln(n))$. (Il est en effet très improbable qu'à chaque fois, on tire au hasard un pivot qui est le plus petit ou le plus grand nombre de la sous-liste que l'on est en train de trier).
3. les algorithmes de type Atlantic City :
   On utilise la fonction `random` pour effectuer un certain nombre de tests alétoires sur un algorithme non probabiliste. Les algorithmes de type Atlantic City donnent une réponse probablement juste avec une complexité probablement efficace.
   Exemple : les tests de primalité permettent de savoir rapidement si un nombre est très probablement premier. On appelle ces nombres les nombres premiers industriels ; ils sont utilisés en cryptologie (ex chiffrage RSA) où l'on a besoin de calculer rapidement des nombres premiers avec une marge de doute infime. Pour les très grands nombres, le test de Miller-Rabin est beaucoup plus rapide que le crible d'Ératosthène mais, dans des cas exceptionnels, il peut déclarer "premier" un nombre qui ne l'est pas (alors que le crible d'Ératosthène n'échoue jamais).