Auxiliar 11

Algoritmos Probabilísticos y Aleatorizados

Profesores: Benjamín Bustos, Gonzalo Navarro
Auxiliares: Sergio Rojas, Pablo Skewes

P1. [P1 - C3 - 2023-1]

El **conteo inexacto** es una técnica aleatorizada para contar con cierto margen de error la cantidad de veces que ocurre un evento, en un entorno donde los números son demasiado grandes como para mantener un contador exacto. La idea es mantener en un contador c sólo el exponente (entero) del valor que queremos contar, es decir, que el evento ha ocurrido unas 2^c veces.

Proponga una estrategia aleatorizada para determinar si aumentar c o no cada vez que ocurre un evento, de modo de garantizar que la cantidad esperada de eventos que han ocurrido es esencialmente 2^c. Analice su estrategia demostrando que tiene esta propiedad. Note que no conoce de antemano cuántas veces ocurrirá el evento.

P2. [P1 - C3 - 2022-2]

Se tienen m conjuntos $S_1, ..., S_m$, de t elementos cada uno, con $m < 2^{t-2}$. Los conjuntos no son disjuntos, habiendo $n \leq mt$ elementos distintos. Se desea colorear esos n elementos de rojo o azul, de modo que ningún S_i quede monocromático (todos los puntos rojos o todos azules).

a) Diseñe un algoritmo de tipo MonteCarlo que, en tiempo $O(mt)$, obtenga un coloreo válido con probabilidad al menos $\frac{1}{2}$.

b) Convierta el algoritmo en uno de tipo Las Vegas y analice su costo esperado y de peor caso.

P3. [P1 - C3 - 2023-2]

Se tienen dos polinomios $P(x)$ y $Q(x)$, de grado n, en $\mathbb{Z}_p$ (es decir, los coeficientes y los argumentos son enteros y se evalúan módulo p), con $n < p$. Se desea saber si $P(x) = Q(x)$ para todo x. Esto no es tan fácil como parece porque pueden estar escritos de distintas formas, por ejemplo $P(x) = (x - 3x^2)(x + 3x^2)$ y $Q(x) = x^2(1 - 9x^2)$, y expandir ambos polinomios puede tomar tiempo exponencial en su largo. En cambio, podemos evaluar un polinomio escrito en largo ℓ en tiempo $O(\ell \log n)$. Supondremos que ℓ es mucho menor que n.

Proponga un algoritmo tipo MonteCarlo one-sided que determine si $P(x) = Q(x)$ en tiempo $O(\ell \log n)$, donde ℓ es el largo de P y Q, y acote su probabilidad de equivocarse en caso de declarar igualdad. Para ello puede serle útil saber que un polinomio de grado n tiene a lo sumo n raíces en $\mathbb{Z}_p$. Luego convierta su algoritmo en uno que se equivoque con una probabilidad dada ε, e indique su costo en términos de ε, ℓ, p y n.