Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2283 - Diseño y Análisis de Algoritmos
Profesor: Nicolás Van Sint Jan
Ayudante: Dante Pinto

Ayudantía 10

Algoritmos aleatorizados y teoría de números

Problema 1: Pequeño teorema de Fermat

Demuestre que si p es un número primo y a es un número natural, entonces:

$$a^p \equiv a \mod p$$

Hint: Demuestre que para x, y enteros y p primo se cumple $(x+y)^p \equiv x^p + y^p \mod p$

Solución: Demostrando primero el hint, tenemos que:

$$\begin{aligned}(x+y)^p &= \sum_{i=0}^{p} \binom{p}{k} x^{p-k} y^k \\ &= \sum_{i=0}^{p} \frac{p!}{k!(p-k)!} x^{p-k} y^k\end{aligned}$$

Luego, si demostramos que para todo k tal que $0 < k < p$ se cumple que $\binom{p}{k}$ es múltiplo de p, habremos demostrado que $\binom{p}{k} \equiv 0 \mod p$ (pues sabemos quae todos estos coeficientes son enteros).

Observando ahora la fracción $\frac{p!}{k!(p-k)!}$, es claro que $p!$ es múltiplo de p, por lo que la única manera de que $\binom{p}{k}$ no sea múltiplo de p es que este factor p se simplifique con el denominador de la fracción. Como tanto k como $p-k$ son estrictamente menores que p y p es un número primo, $k!$ y $(p-k)!$ no compartirán factores con p (además del 1), lo que significa entonces que $\binom{p}{k} \equiv 0 \mod p$.

$$\begin{aligned}\Rightarrow (x+y)^p &\equiv \sum_{i=0}^{p} \binom{p}{k} x^{p-k} y^k \mod p \\ &\equiv \binom{p}{0} x^p y^0 + \binom{p}{p} x^0 y^p \mod p \\ &\equiv x^p + y^p \mod p\end{aligned}$$

Teniendo lo anterior, podemos demostrar el teorema usando inducción. Como caso base tenemos:

$$0^p = 0 \equiv 0 \mod p$$

Luego suponemos que:

$$a^p \equiv a \mod p$$

Y finalmente desarrollamos:

$$\begin{aligned}
(a+1)^p &\equiv a^p + 1^p \mod p && \text{(Hint)}\\
&\equiv a^p + 1 \mod p\\
&\equiv a + 1 \mod p && \text{(Hipótesis de inducción)}
\end{aligned}$$

Por lo tanto $a^p \equiv a \mod p$ para todo p primo y a natural.

Problema 2: Comunicación aleatorizada

Una persona A desea enviarle un mensaje M, codificado como un string binario de m bits, pero por problemas de conexión solo puede enviar $n < m$ bits.

Para resolver lo anterior, A define $p = \frac{M}{2^m}$ y envía n bits, eligiendo un 1 con probabilidad p y un 0 con probabilidad $1 - p$, de manera independiente para cada bit.

1. Entregue un algoritmo que decodifica el mensaje de A a partir de los n bits y el largo m del mensaje.

   Solución: Los bits recibidos formarán un string binario $b = b_1 ... b_n$ y, utilizando los bits, podemos intentar estimar la probabilidad p que eligió A.

   $$\overline{b} = \frac{1}{n}\sum_{i=1}^{n} b_i$$

   Con esta probabilidad estimada podemos multiplicarla por 2^m y luego redondear este valor (al entero más cercano) encontrando el mensaje decodificado. De esta forma calculamos:

   $$M_0 = \overline{b} \cdot 2^m$$
   $$M' = \texttt{round}(M_0)$$

2. Acote la probabilidad de que el mensaje decodificado sea incorrecto.

   Solución: Usando el método anterior, claramente fallaremos cuando $M' \leq M$, lo que puede ocurrir por dos razones, la primera es que el mensaje M_0 era similar a M, pero al redondear nos acercamos a un entero distinto de M y la segunda es que el mensaje calculado, M_0, simplemente no se parecía en nada al original. Estos dos casos pueden representarse por el mismo evento, la distancia entre M_0 y M es mayor o igual a 0.5.

   $$\begin{aligned}
   Pr[error] &= Pr\left[|M_0 - M| \geq \frac{1}{2}\right]\\
   &= Pr\left[|\overline{b} \cdot 2^m - p \cdot 2^m| \geq \frac{1}{2}\right]\\
   &= Pr\left[|\overline{b} - p| \geq \frac{1}{2^{m+1}}\right]
   \end{aligned}$$

   La probabilidad anterior es bastante similar a la cota de Chebychev $\left(Pr[|X - E[X]| \geq a] \leq \frac{Var[X]}{a^2}\right)$, por lo que si encontramos la esperanza de nuestra variable aleatoria, $\overline{b}$, podremos aplicar la cota.

$$
\begin{aligned}
E[\bar{b}] &= E\left[\frac{1}{n}\sum_{i=0}^{n} b_i\right] \\
&= \frac{1}{n}\sum_{i=0}^{n} E\left[b_i\right] \\
&= \frac{1}{n}\sum_{i=0}^{n} p \\
&= p
\end{aligned}
$$

$$
\begin{aligned}
Pr[error] &= Pr\left[|\bar{b} - p| \geq \frac{1}{2^{m+1}}\right] \\
&= Pr\left[|\bar{b} - E[\bar{b}]| \geq \frac{1}{2^{m+1}}\right] \\
&\leq Var[\bar{b}] \cdot 2^{2m+2}
\end{aligned}
$$

Finalmente, calculando la varianza de $\bar{b}$ encontramos:

$$
\begin{aligned}
Var[\bar{b}] &= \frac{1}{n^2}\sum_{i=0}^{n} Var[b_i] \\
&= \frac{1}{n^2} \cdot n \cdot Var[b_i] \\
&= \frac{1}{n}(E[b_i^2] - E[b_i]^2) \\
&= \frac{1}{n}(E[b_i] - E[b_i]^2) \\
&= \frac{1}{n}(p - p^2) \\
\therefore Pr[error] &\leq \frac{p(1-p)}{n} \cdot 2^{2m+2}
\end{aligned}
$$

3. ¿Cuántos bits debe enviar A para que la probabilidad de error sea menor a ε? ¿Es eficiente la solución de A?

   Solución: Para calcular esto simplemente necesitamos reemplazar la probabilidad de error por ε y despejar:

$$
\begin{aligned}
\varepsilon &\leq \frac{p(1-p)}{n} \cdot 2^{2m+2} \\
\Rightarrow n &\geq \frac{p(1-p)}{\varepsilon} \cdot 2^{2m+2} \\
&\geq \frac{1}{4\varepsilon} \cdot 2^{2m+2} \\
&= \frac{2^{2m}}{\varepsilon}
\end{aligned}
$$

   La solución **no** es eficiente, pues, por ejemplo, para conseguir una probabilidad de error menor o igual a $\frac{1}{2}$ necesitaremos enviar 2^{2m+1} bits, que claramente es mayor que los m bits correspondientes a mandar el mensaje completo.