Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2283 - Diseño y Análisis de Algoritmos - 2$^{\text{do}}$ semestre 2022

PAUTA INTERROGACIÓN 2

Pregunta 1

Sea $a \in \mathbb{Z} - \{0\}$ y $n \in \mathbb{N} - \{0\}$. Demuestre que a tiene inverso en módulo n si, y solo si, $\gcd(a, n) = 1$

Solución. Primero suponga que b es inverso de a en módulo n,

$$a \cdot b \equiv 1 \pmod{n}.$$

Podemos decir entonces que $a \cdot b = \alpha \cdot n + 1$. Reescribiendo,

$$1 = a \cdot b - \alpha \cdot n$$

Se concluye entonces que para cualquier c que divida a a y n al mismo tiempo, i.e. $c \mid a$ y $c \mid n$, debe ser cierto que $c \mid 1$. Entonces necesariamente $c = 1$ y por lo tanto $\gcd(a, n) = 1$.

Ahora, suponga que $\gcd(a, n) = 1$. Por la identidad de Bézout existen $s, t \in \mathbb{Z}$ tales que:

$$1 = s \cdot n + t \cdot a$$

luego trabajando con congruencias modulares tenemos que

$$1 \equiv s \cdot n + t \cdot a \equiv t \cdot a \pmod{n}$$

lo que quiere decir que t es el inverso de a en módulo n.

Rúbrica. Dado lo anterior la atribución de puntaje es la siguiente:

(**1 Punto**) El caso directo: notar que $1 = a \cdot b - \alpha \cdot n$.
(**2 Puntos**) El caso directo: argumentar que entonces $\gcd(a, n) = 1$.
(**1 Punto**) El caso converso: notar que se puede utilizar la identidad de Bézout.
(**2 Puntos**) El caso converso: argumentar que t resulta ser el inverso de a en módulo n.

Pregunta 2

Sea **Moneda**() un procedimiento que retorna 1 con probabilidad p y 0 con probabilidad $1 - p$, donde $p \in [0, 1]$. Se define el procedimiento **EstimarMoneda**() como el siguiente algoritmo:

EstimarMoneda(n)
  $s := 0$
  for $i := 1$ **to** n **do**
    $s := s +$ **Moneda**()
  return $\frac{s}{n}$

1. Dado $\epsilon \in (0, 1)$, demuestre que:

$$\Pr(|\textbf{EstimarMoneda}(n) - p| < \epsilon) \geq 1 - \frac{p(1-p)}{n\epsilon^2}$$

2. Calcule un valor de n tal que para todo $p \in [0,1]$, el valor retornado por **EstimarMoneda**(n) tiene un error de estimación de p de a lo más 1% con una probabilidad mayor o igual a $\frac{999}{1000}$. Es decir, el valor de n encontrado debe satisfacer:

$$\forall p \in [0,1].\ \Pr\left(|\textbf{EstimarMoneda}(n) - p| < \frac{1}{100}\right) \geq \frac{999}{1000}$$

Hint: Puede utilizar el resultado del ítem anterior aunque no lo haya demostrado.

Solución. A continuación se muestra una posible demostración para cada inciso:

1. Primero se debe considerar que el resultado de **EstimarMoneda**(n) es una variable aleatoria (llamémosla X) que puede ser escrita en términos de un promedio de variables aleatorias $X_i \sim \text{Bernoulli}(p)$ para $1 \leq i \leq n$ tal que

$$X_i = \begin{cases} 1 & \text{si la } i\text{-ésima llamada a } \textbf{Moneda}() \text{ retorna } 1 \\ 0 & \text{si la } i\text{-ésima llamada a } \textbf{Moneda}() \text{ retorna } 0 \end{cases}$$

Así entonces

$$X = \frac{1}{n}\sum_{i=1}^{n} X_i$$

La esperanza de X es entonces:

$$\mathrm{E}(X) = \mathrm{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}(X_i) = \frac{1}{n}\sum_{i=1}^{n} p = \frac{1}{n}\cdot np = p$$

Mientras que la varianza de X será (dado que X_i es independiente de X_j para todo $1 \leq i < j \leq n$):

$$\mathrm{Var}(X) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}(X_i) = \frac{1}{n^2}\sum_{i=1}^{n} p(1-p) = \frac{1}{n^2}\cdot np(1-p) = \frac{p(1-p)}{n}$$

Si reemplazamos en la desigualdad de Chebyshev tomando como constante positiva a ϵ obtenemos:

$$\Pr(|X - \mathrm{E}(X)| \geq a) \leq \frac{\mathrm{Var}(X)}{a^2} \qquad \Rightarrow \qquad \begin{aligned} \Pr(|X - p| \geq \epsilon) &\leq \frac{p(1-p)}{n\epsilon^2} \\ 1 - \Pr(|X - p| \geq \epsilon) &\geq 1 - \frac{p(1-p)}{n\epsilon^2} \\ \Pr(|X - p| < \epsilon) &\geq 1 - \frac{p(1-p)}{n\epsilon^2} \end{aligned}$$

Que es justamente lo que se pedía demostrar.

2. Para encontrar el valor de n primero se reemplaza $\epsilon = \frac{1}{100}$ en el resultado anterior:

$$\Pr\left(|X - p| < \frac{1}{100}\right) \geq 1 - \frac{p(1-p)}{n}\cdot 100^2$$

Es decir, estamos buscando resolver la siguiente inecuación de incógnita n para $p \in [0,1]$:

$$\begin{aligned} 1 - \frac{p(1-p)}{n}\cdot 100^2 &\geq \frac{999}{1000} \\ \frac{1}{1000} &\geq \frac{p(1-p)}{n}\cdot 100^2 \\ n &\geq p(1-p)\cdot 10^7 \end{aligned}$$

notando que $p(1-p)$ posee un máximo en $p = \frac{1}{2}$ se concluye que

$$n \geq \frac{1}{2}\left(1 - \frac{1}{2}\right)\cdot 10^7 = \frac{1}{2}\cdot\frac{1}{2}\cdot 10^7 = \frac{10^7}{4} = 2.500.000$$

luego cualquier $n \geq 2.500.000$ cumple con lo pedido.

Rúbrica. Dado lo anterior la atribución de puntaje es la siguiente:

En ítem 1.

(**1 Punto**) Por definir la v.a. X en términos de un promedio de variables $X_i \sim \text{Bernoulli}(p)$.

(**1 Punto**) Por encontrar $\text{E}(X)$ y $\text{Var}(X)$ correctamente.

(**1 Punto**) Por usar la desigualdad de Chebyshev para obtener el resultado pedido.

En ítem 2.

(**1 Punto**) Por reemplazar el resultado anterior con $\epsilon = \frac{1}{100}$ y enunciar la inecuación.

(**2 Puntos**) Por notar que $p(1-p)$ tiene un máximo en $p \in [0,1]$ y logrando encontrar el n pedido.

Pregunta 3

Considere el siguiente algoritmo de Las Vegas que realiza una búsqueda de un elemento x en una lista desordenada $L[1 \dots n]$ que NO tiene elementos repetidos.

BúsquedaAleatoria$(L[1 \dots n], x)$
  escoja al azar y con distribución uniforme un número i
    del conjunto de números naturales $\{1, ..., n\}$
  if $L[i] = x$
    return i
  else
    return *sin_resultado*

1. Encuentre el número esperado de veces que se debería ejecutar **BúsquedaAleatoria** en promedio hasta obtener una respuesta correcta en función de n. Puede asumir que el elemento x siempre estará presente en $L[1 \dots n]$.

2. Construya un algoritmo de Monte Carlo a partir del algoritmo **BúsquedaAleatoria** que resuelva el mismo problema y cuya probabilidad de error sea a lo más $\frac{1}{2}$. Explique por qué su algoritmo cumple con lo pedido.

   Hint: La desigualdad
$$\left(1 - \frac{1}{n}\right)^{an} < e^{-a}$$
   que se cumple para $a > 0$ y $n \geq 1$ puede resultar útil.

3. Demuestre un caso general del punto anterior. Sea $\mathcal{A}$ un algoritmo de Las Vegas tal que el tiempo esperado total de un algoritmo que ejecuta $\mathcal{A}$ hasta obtener una respuesta correcta es $T(n)$. Demuestre que para cada $c \geq 0$ siempre existe un algoritmo $\mathcal{B}$ de Monte Carlo que computa el mismo problema que $\mathcal{A}$ en tiempo $c \cdot T(n)$ y cuya probabilidad de error es a lo más $1/c$.

Solución. A continuación se muestra una posible demostración para cada inciso:

1. Sea X la variable aleatoria que describe la cantidad de veces que se ejecuta **BúsquedaAleatoria**(L, x) hasta obtener una respuesta correcta. Es claro que el dominio de X es el conjunto $\{1, 2, 3, \dots\}$. Se está pidiendo la esperanza de X:
$$\text{E}(X) = \sum_{i=1}^{\infty} i \cdot \Pr(X = i)$$
   Nótese que $\Pr(X = i)$ implica que en las $(i-1)$-primeras ejecuciones de **BúsquedaAleatoria**(L, x) no se obtuvo respuesta y en la i-ésima ejecución sí se obtuvo respuesta. Dado que la probabilidad de obtener

una respuesta es $\frac{1}{n}$ luego:

$$\begin{aligned}
\mathrm{E}(X) &= \sum_{i=1}^{\infty} i \cdot \left(1 - \frac{1}{n}\right)^{i-1} \frac{1}{n} \\
&= \frac{1}{n} \sum_{i=1}^{\infty} i \cdot \left(\frac{n-1}{n}\right)^{i-1}
\end{aligned}$$

Se puede dejar expresado este resultado y se considerará como correcto. Sin embargo esta suma infinita converge a n:

$$\mathrm{E}(X) = n$$

También se podría haber argumentado simplemente que la variable X es tal que $X \sim \text{Geométrica}\left(\frac{1}{n}\right)$, y por lo tanto $\mathrm{E}(X) = n$.

2. Un posible algoritmo de Monte Carlo que resuelve el problema consiste en lo siguiente:

   BúsquedaMonteCarlo$(L[1 \dots n], x)$
   $\quad f := 0$
   $\quad$ **for** $r := 1$ **to** n **do**
   $\quad\quad i :=$ **BúsquedaAleatoria**(L, x)
   $\quad\quad$ **if** $i \neq \textit{sin\_resultado}$ **then**
   $\quad\quad\quad f := i$
   $\quad$ **return** f

   La probabilidad de error del algoritmo anterior corresponde a que en ninguno de los n intentos de llamar a **BúsquedaAleatoria**(L, x) se llegue a un resultado. Es decir:

   $$\Pr(\textbf{BúsquedaMonteCarlo}(L[1 \dots n], x) \text{ se equivoca}) = \left(1 - \frac{1}{n}\right)^n < \frac{1}{e} < \frac{1}{2}$$

3. Sea $\mathcal{A}$ el algoritmo de Las Vegas. Decimos que $T(n)$ es el tiempo promedio de la ejecución repetida de $\mathcal{A}$ hasta encontrar una respuesta. Sea X la variable aleatoria que representa el tiempo de ejecutar $\mathcal{A}$ repetidas veces hasta obtener una respuesta. Entonces,

   $$\mathrm{E}(X) = T(n)$$

   Definamos $\mathcal{B}$ como el algoritmo que hace lo siguiente:

   1. Mantiene un contador de tiempo que se inicializa en cero y por cada unidad mínima de tiempo que pasa aumenta c en uno.
   2. Hace llamadas repetidas a $\mathcal{A}$ y guarda su respuesta si alguna vez entrega un resultado.
   3. Cuando el marcador de tiempo llega a $c \cdot T(n)$ el algoritmo deja de hacer llamadas a $\mathcal{A}$.
   4. Si $\mathcal{A}$ entregó respuesta durante alguna ejecución, se entrega esa respuesta. Si no, se retorna cualquier cosa (cero por ejemplo).

   Es claro que $\mathcal{B}$ siempre termina y entrega respuesta en tiempo $c \cdot T(n)$ con posibilidad de equivocarse, por lo que efectivamente se trata de un algoritmo de Monte Carlo.

   La probabilidad de que $\mathcal{B}$ se equivoque es la misma de que la ejecución repetida de $\mathcal{A}$ no encuentre respuesta en tiempo $c \cdot T(n)$. Es decir,

   $$\Pr\left(\mathcal{B} \text{ se equivoque}\right) = \Pr(X \geq c \cdot T(n))$$

   Nos queda demostrar que esta probabilidad está acotada por $\frac{1}{c}$. Para esto podemos utilizar la desigualdad de Markov. Dado que X es una variable aleatoria positiva, lo siguiente se cumple para $a > 0$:

   $$\Pr(X \geq a) \leq \frac{\mathrm{E}(X)}{a} = \frac{T(n)}{a}$$

Luego si reemplazamos $a = c \cdot T(n)$:

$$\Pr(X \geq c \cdot T(n)) \leq \frac{T(n)}{c \cdot T(n)} = \frac{1}{c}$$

con lo que se concluye la demostración.

Rúbrica. Dado lo anterior la atribución de puntaje es la siguiente:

En ítem 1.

(**1 Punto**) Por definir la v.a. X y encontrar correctamente $\Pr(X = i)$ para algún $i \in \mathbb{N}$.

(**1 Punto**) Por encontrar una expresión correcta para $\mathrm{E}(X)$.

En ítem 2.

(**1 Punto**) Por definir un algoritmo de Monte Carlo acorde a lo pedido.

(**1 Punto**) Por demostrar que la probabilidad de error del algoritmo entregado está acotada por $\frac{1}{2}$.

En ítem 3.

(**1 Punto**) Por definir correctamente un algoritmo $\mathcal{B}$ de Monte Carlo que cumpla con lo pedido.

(**1 Punto**) Por demostrar que la prob. de error de $\mathcal{B}$ es a lo más $\frac{1}{c}$ usando la desigualdad de Markov.