Ma. Isabel Ortiz Naranjo

Carne: 18176

Tarea 4 – Análisis de algoritmos

1. Describa una implementación de un generador de números enteros aleatorios dentro del intervalo $[a, b]$, llamada `random(a,b)`. Su generador debe estar basado en llamadas a otro generador de bits aleatorios ($p(0) = p(1) = \frac{1}{2}$, donde p es la función de probabilidad) llamado `random(0,1)`. Luego, calcule el tiempo de ejecución esperado de su algoritmo. ***Hint***: tome en cuenta que la probabilidad para cada número desde a hasta b debe ser la misma y no olvide que las probabilidades deben sumar 1. Investigue la distribución geométrica.

Notas

Haciendo un supuesto que al generar un bit en un tiempo considerablemente constante, se obtiene el siguiente algoritmo: $\Theta(\log 2(b-a))$ que este produce un valor aleatorio.

Entonces el tiempo de ejecución es:

$$O(2^{\lceil \log 2(b-a) \rceil} / b-a * \log 2(b-a))$$

Sin embargo, $\lceil \log 2(b-a) \rceil < \log 2(b-a)+1$, entonces se puede decir que $2^{\lceil \log 2(b-a) \rceil} / b-a < 2(b-a)/b-a = 2 = O(1)$.

Quiere decir que el tiempo de ejución final sería $O(\log 2(b-a))$.

2. Dado un generador de bits llamado `biased-random()` que produce 1 con probabilidad p y 0 con probabilidad $1 - p$ (donde $0 < p < 1$), diseñe un algoritmo que use `biased-random()` para generar 1 o 0, ambos con probabilidad $\frac{1}{2}$. Calcule el tiempo de ejecución esperado. ***Hint***: ¿cuál es la probabilidad de que dos ejecuciones seguidas de `biased-random()` den el mismo resultado? Observe que las ejecuciones de `biased-random()` son independientes entre sí.

De la suma de las probabilidades tenemos:

$p(1-p) + (1-p)p = 2p(1-p)$

Entonces, el número esperado de ejuciones antes de parar es de $12p(1-p)$, esto determina el tiempo de ejecución esperado del algoritmo al multiplicarse por el tiempo que tome biased-random().

3. ¿Cuáles son las probabilidades de obtener el *best-case scenario* (contratar sólo una vez) y el *worst-case scenario* (contratar n veces) si la distribución del *input* del *hiring problem* es aleatoria uniforme? ***Hint***: recuerde que el primer entrevistado siempre es contratado, independientemente de cómo vengan ordenados los que le siguen.

* Recordar que primer entrevistado siempre es contratado.
n candidatos, calificación $(1-n)$

° Hay $(n-1)!$ permutaciones. Quiere decir que la probabilidad una vez es $\frac{(n-1)!}{n!} = \frac{1}{n}$. y contratar n veces es $\frac{1}{n!}$

4. Use variables aleatorias para calcular la suma esperada de n dados lanzados a la vez.

variable aleatoria → a

° Sumatoria = $a = \sum_{i=1}^{n} d_i$

$E[a] = E[a_{i=1}^{n} d_i]$

Entonces: la probabilidad de que caiga en cada cara es de $\frac{1}{6}$.

$E[d_i] = \sum_j^n \times \frac{1}{6} = \sum_{i=1}^{n} \frac{21n}{6}$

5. Suponga una lista A con una permutación aleatoria de los números $[1..n]$. Una **inversión** es una pareja (i, j) donde $i < j$ pero $A[i] > A[j]$. Use variables aleatorias indicadoras para calcular el número esperado de inversiones en A. ***Hint***: ¿cuál es la probabilidad de que un par de posiciones i, j cualesquiera en una permutación aleatoria de n números cumplan con que $A[i] > A[j]$?

Suponer una lista A con una permutación aleatoria de los números [1...n]

° Problema conteo

° Variable indicadora ¿?

$X_{i,j} = 1$ si hay inversión

Entonces:

$$\Sigma[X] = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} E[X_{i,j}] = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} P(\lambda)$$

\# de permutaciones: $\binom{n}{2} \times (n-2)!$

Entonces: (tomando en cuenta las permutaciones)

$$\Sigma[X] = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \frac{1}{2} = \frac{1}{2} * \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} 1 = \frac{1}{2} * \binom{n}{2} = \frac{n(n-1)}{4}$$

6. Descargue el *paper "Sorting the Slow Way: An Analysis of Perversely Awful Randomized Sorting Algorithms"*. Luego, responda las siguientes preguntas:
   a. Explique la fórmula usada para calcular $P[I_k]$ en la prueba del teorema 2.
   b. En la prueba del teorema 2, ¿por qué $E[C] = \sum_{k>0} P[I_k]$?
   c. En la sección 2.3, ¿por qué la variable aleatoria I que cuenta el número de iteraciones del algoritmo tiene distribución geométrica?

a.) $P[I_k] \rightarrow$ ¿Fórmula usada para calcular?

Combinaciones $k \rightarrow \binom{n}{k}$

números $\rightarrow (n-k)$

Total de posibilidades es $\rightarrow \binom{n}{k} * (n-k)!$

Sería $\frac{1}{k!}$ Respuesta

b.) ¿Por qué $E[C] = \sum_{k>0} P[I_k]$?

$I_k = 1$ se suman las probabilides

$P[C=j]$ para $j = k, k+1, k+2 \ldots$

$$P[C \geq k] = \sum_{j \geq k} P[C=j]$$

Entonces:

$$\sum_{k>0} P[C \geq k] = \sum_{k>0} P[I_k]$$

c. ¿Por qué la variable aleatoria I que cuenta el número de iteraciones del algoritmo tiene distribución geométrica?

→ Esta variable I cumple con la definición de la distribución geométrica, ya que cada intento es independiente del anterior por permutar el arreglo completamente al azar.

7. Descargue el *paper* *"Fun-sort – or the chaos of unordered binary search"*. Luego responda las siguientes preguntas:
   a. El algoritmo *guess-sort* presenta una mejora con respecto a *bozo-sort*$^{+}_{opt}$ (sección 3 del *paper* del inciso anterior). ¿Cuál es la diferencia que permite esta mejora?
   b. En las conclusiones, una pregunta sugiere que el tiempo de ejecución de *Fun-sort* en el *average-case* ha de ser más o menos rápido gracias al teorema 6. Si las condiciones del teorema 6 se cumplen, ¿cuál sería el tiempo de ejecución esperado?

a. Guess-sort

→ No se encarga que arreglo esté ordenado luego de cada intercambio de posiciones

→ Elige 2 posiciones al azar, determinando si son inversión, si lo son hace intercambio.

b. Esta búsqueda termina hasta que colapsan los límites de búsqueda.

→ Se toma un tiempo $O(\log_2 n)$