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) = ½, 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 tempo considerablemente constante, se obtiene el piquiente algoritmo: $O(\log 2(b-a))$ que este produce un valor aleatorios.
Entonces el tempo de ejecución es:
D(2 log 2 (b-a) 16-9 * log 2 (b-a))
Sin embarge, [leg2 (b-a)] < leg2 (b-a) + 1, entonces se puede deur que 2 (log2 (b-a) b-a < 2 (b-a) b-9 = 2 - 0(1).  Quiere dur que el tiempo de ejución final pería 0 (log2 (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

Pe la auma de las probabilidades tenemos:

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

Entonas, el número esperado de ejuciones antes de
parar es de 12 p(1-p), esto determinas el tiempo
de ejecución esperado del algoritmo al nultiplicarse
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 entreustado siempre es contratado.

n candidatos, calificación (1-n)

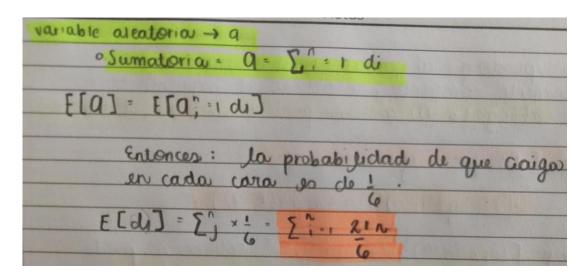
\* Hay (n-1)! permutaciones Quiere decir que la propabilidad

una vez es (n-1)!=! y contratar n veces es !

n!

n!

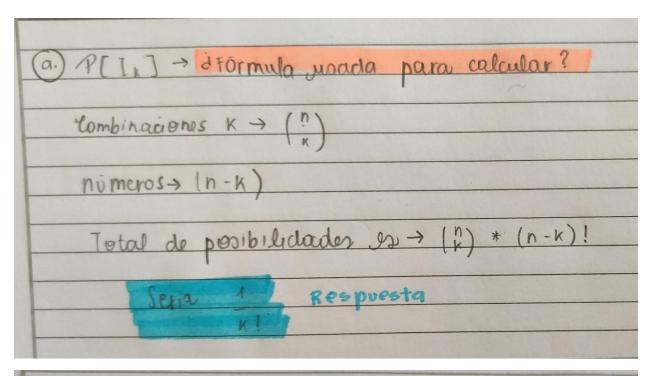
4. Use variables aleatorias para calcular la suma esperada de n dados lanzados a la vez.



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]?

Problema content  Ovariable indicadora i?  Xi, j = 1 pi Jay inversion  Entonas:  1-1 h+1  1-1 h+1  1-1 h+1  1-1 h+1  1-1 h+1	
Entonas:  n-1 h+1  n-1 n+1  P[V] - 5 5 F[V : 7 = 5 P/2)	
$E[X] = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} \frac{\sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} P(x)}{\sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} P(x)}$ $= \frac{\sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} P(x)}{\sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} P(x)}$ $= \frac{\sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} P(x)}{\sum_{i=0}^{n-1} \sum_{j=i+1}^{n+1} P(x)}$	
Entonces: (tomando en cuenta las permutaciones) $ \begin{array}{cccccccccccccccccccccccccccccccccc$	

- 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?



b. ¿ por que E[C] = Ex 20 P[Ik]?
In=1 pe puman las probabilides
P[C=j] para $j=K$ , $K+1$ , $K+2$
P[C \ K] = \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
Enlonces:
EPECSKJ = E * EIKJ

c.) à l'or que la variable akatoria I que auntai al número de iteraciones del algoritmo tiene distribución geométrica?

Testa variable I cumple con la definición de la distribución geométrica, ya que cada intento so independente del anterior por permutar el arreguo 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<sup>+</sup>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
lugo de cada intercambio de posiciones  Tige 2 posiciones al azar determinanto
- Tige 2 posiciones al azar, determinanto
intercambio.
intercambio.
(b) Esta hisando termino hosto an colonian
los limites de hisaveda.
b.) téta bisqueda termina hasta que colapsan  Jos limites de bisqueda  → Se toma un tiempo 0 (log n)