Universidad de Chile
Departamento de Ciencias de la Computación
CC4102 - Diseño y Análisis de Algoritmos

Auxiliar 8 - "Algoritmos Probabilistas y Aleatorizados"

Profesor: Gonzalo Navarro
Auxiliar: ~~Manuel~~ Ariel Cáceres Reyes

30 de Octubre del 2017

P1. Mejor precio de compra

Intentamos comprar un artículo en una feria de artesanías en la cual n vendedores lo ofrecen a distintos precios ya fijados, pero desconocidos. Podemos ir a preguntarle a un vendedor a que precio vende el artículo, sin embargo, los vendedores se ofenden fácilmente, por lo que no negociarían con nosotros si rechazamos su oferta.
Diseñe una técnica aleatorizada que con probabilidad al menos 1/4 consiga comprar el artículo al mejor precio posible.

P2. Prender servidores

Todos los días se necesitan prender n servidores. Para prender un servidor solo es necesario enviar una señal a ese servidor quien, luego de recibirla, tarda k segundos en prenderse. Un servidor prendido que recibe esta señal la ignora. Por otro lado, no se permite que 2 servidores se estén prendiendo al mismo tiempo.
Para automatizar esta tarea dispone de un timer y un generador aleatorio de señales, que al ser energizado envía una señal al azar a alguno de los n servidores.

a) Diseñe un sistema tipo "Las Vegas", calcule su tiempo esperado.
b) Diseñe un sistema tipo "Montecarlo" cuya probabilidad de fracaso sea tan pequeña como se desee(con una penalización en tiempo de ejecución).

P3. 3 Mayoritario

Considere un árbol de altura h perfectamente balanceado, donde cada nodo interno tiene 3 hijos, por lo que hay $n = 3^h$ hojas. Cada hoja tiene un valor booleano, 0 ó 1. El valor booleano de cada nodo interno se calcula como el mayoritario entre sus 3 hijos.

a) Muestre que cualquier algoritmo determinístico necesita en el peor caso examinar las $n = 3^h$ hojas para encontrar el valor de la raíz.
b) Considere un algoritmo aleatorizado que elige dos hijos al azar, los evalúa recursivamente, y si dan el mismo valor evita calcular el tercero, de otro modo calcula el tercero para responder.
Demuestre que el número esperado de hijos analizados por cada nodo es 8/3.
c) Analice el costo esperado del algoritmo aleatorizado y muestre que es $o(n)$.

P4. Polinomios

Dados 3 polinomios en una variable p, q de grado n y r de grado $2n$. Muestre un algoritmo probabilístico que tome tiempo $\mathcal{O}(n)$ en determinar si $pq = r$.

"Carry out a random act of kindness, with no expectation of reward, safe in the knowledge that one day someone might do the same for you."
Princesa Diana

Soluciones

P1. Si vamos a los vendedores en algún orden aleatorio, y rechazamos la primera mitad de ofertas, guardando el precio más bajo que nos ofrecieron y luego aceptamos la primera oferta menor al mínimo que tenemos guardado alcanzamos esa probabilidad.
Para mostrar que la probabilidad de cumplir nuestro objetivo es al menos $1/4$ veamos que elegir un orden aleatorio en el que visitar es análogo a elegir una permutación al azar entre los vendedores. Un evento favorable para el algoritmo es el que el mínimo se encuentre en la segunda mitad y el segundo mínimo en la primera, la probabilidad de este evento será entonces menor a la probabilidad de éxito de nuestro algoritmo y esta probabilidad es $\frac{(n-2)!\cdot n/2\cdot n/2}{n!} \geq 1/4$.

P2. a) Con el timer le energizo el generador aleatorio cada k segundos infinitamente.
Definimos la variable aleatoria Y_i como el número de señales mandadas hasta que se prenda el i-ésimo servidor (y por convención $Y_0 = 0$). El valor que nos interesa calcular entonces es $\mathbb{E}(kY_n)$.
Para esto definamos $X_i = Y_{i+1} - Y_i$ interpretado como el número de señales mandadas entre que se prendió el i-ésimo servidor y se prende el siguiente. Notemos que $X_i \sim Geom\left(\frac{n-i}{n}\right)$.
Por lo tanto:

$$\begin{aligned}\mathbb{E}(Y_n) &= \mathbb{E}(X_0 + X_1 + \ldots + X_{n-1}) \\ &= \sum \mathbb{E}(X_i) \\ &= \sum_{i=0}^{n-1} \frac{n}{n-i} \\ &= n \sum_{i=1}^{n} \frac{1}{y} \\ &= nH_n\end{aligned}$$

que para al menos 2 servidores es $\leq 2nlgn$.

b) Con el timer le energizo el generador aleatorio cada k segundos, $2nH_n$ veces.
Por la desigualdad de Markov se cumple que:

$$\mathbb{P}(fracaso) =$$

$$\mathbb{P}(Y_n \geq 2nH_n) \leq \frac{nH_n}{2nH_n} = 1/2$$

Finalmente podemos hacer repetir este experimento i veces para obtener una probabilidad de fracaso menor o igual a $\frac{1}{2^i}$.

P3. a) Sea algoritmo A que resuelve el problema. Mantendremos un adversario 😈que a cada pregunta en una hoja responde:

"Carry out a random act of kindness, with no expectation of reward, safe in the knowledge that one day someone might do the same for you."
Princesa Diana

- 1 si es el primer hijo preguntado.
- 0 si es el segundo hijo preguntado.
- Si es el tercer hijo preguntado, reconoce el ancestro más bajo que tiene alguna hoja no preguntada en su subárbol. En caso que los otros dos hijos tengan hojas no preguntadas, el adversario responde 1. En caso que solo uno de sus hijo tenga hojas no preguntadas respondo 0. Finalmente si no existe tal ancestro respondo 1.

Notemos que si no existe el ancestro del punto 3 significa que ya han sido consultadas todas las hojas. Por otro lado, en todo momento, si se han consultado menos de 3^h hojas que el adversario puede completar con 0s o 1s a su gusto el resto de las hojas, obteniendo resultados de evaluación diferentes y por lo tanto haciendo que el algoritmo se equivoque frente a cualquier respuesta que de.

b) No importa cual sea el input, en un nodo siempre se cumple que al menos dos de sus hijos deben evaluar al mismo valor y por lo tanto, con probabilidad $\leq 1/3$ el algoritmo elige el nodo diferente, solo teniendo que analizar 2 nodos y con probabilidad $\leq 2/3$ debe analizar 3. Con esto el número esperado de hijos a analizar será $\leq 1/3 \cdot 2 + 2/3 \cdot 3 = 8/3$.

c) Si definimos el costo aleatorizado para evaluar un árbol de altura h tendremos que :

$$T(h) \leq \frac{1}{3} \cdot 2 \cdot T(h-1) + \frac{2}{3} \cdot 3 \cdot T(h-1) = \frac{8}{3} T(h)$$

Y por lo tanto $T(h) \leq \left(\frac{8}{3}\right)^h$, que significa $T(h) = \mathcal{O}\left(n^{\log_3 \frac{8}{3}}\right) = o(n)$.

P4. Elegimos uniformemente $x \in \{0, \ldots, 4n+1\}$ y luego respondemos `true` si $p(x) \cdot q(x) = r(x)$ y `false` en otro caso.

- En el caso de que la igualdad sea correcta, el algoritmo responde `true` de manera correcta.
- Sin embargo, cuando la igualdad es falsa ($pq \neq r$) el algoritmo podría responder erróneamente `true` en caso que $p(x) \cdot q(x) = r(x)$ o equivalentemente cuando el polinomio $g = pq - r$ se hace cero, i.e. cuando x es cero de g.
  Veamos que g es de grado a lo más $2n+1$ por lo que tiene a lo más $2n+1$ raíces, y luego la probabilidad que el algoritmo se equivoque es:

$$\mathbb{P}(x \; es \; raiz \; de \; g) \leq \frac{2n+1}{4n+2} = \frac{1}{2}$$

"Carry out a random act of kindness, with no expectation of reward, safe in the knowledge that one day someone might do the same for you."
Princesa Diana