

IIC2283 — Diseño y Análisis de Algoritmos

Ayudantía 8

Problema 1

Se considera un algoritmo arbitrario A , que funciona en tiempo promedio n^2 con entradas de n bits elegidas de forma uniforme.

Parte 1 Describa una cota superior al costo en el peor caso, considerando entradas del mismo largo.

Solución Una solución se basa en la desigualdad de Markov. Si X es la variable aleatoria correspondiente al tiempo de ejecución, entonces se busca t tal que $\Pr[X \geq t] = 0$. Como hay 2^n casos posibles, que se eligen de forma uniforme, entonces basta que $\Pr[X \geq t] < 1/2^n$. Esta desigualdad se cumple cuando $n^2/t < 1/2^n$, por que en ese caso:

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t} = \frac{n^2}{t} < \frac{1}{2^n}$$

Para que se cumpla la última desigualdad, se puede considerar $t > n^2 \cdot 2^n$.

Parte 2 Generalice el resultado anterior a algoritmos de tiempo promedio T , con N entradas posibles.

Idea de solución Similar al caso anterior:

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t} = \frac{T}{t} < \frac{1}{N}$$

que sucede con $t > TN$.

Problema 2

A necesita enviar un mensaje de m bits a B , quien conoce el largo del mensaje. Para esto, A envía múltiples mensajes de un bit, pero no mantiene información sobre los mensajes enviados, es decir no recuerda la cantidad de mensajes enviados, ni el contenido de estos. Adicionalmente, B no puede enviar mensajes.

Proponga un protocolo aleatorizado donde B pueda reconstruir el mensaje original con probabilidad de error a lo más $\epsilon > 0$.

Idea de solución El mensaje se puede interpretar como la representación binaria de un número entre 0 y $2^m - 1$. Luego, si ese número se denota como M , se puede definir $p = M/(2^m)$ entre 0 y 1.

El protocolo consiste en que A envía 1 con probabilidad p , y 0 con probabilidad $(1 - p)$ todas las veces necesarias. B calcula el promedio de los primeros t mensajes, para un t suficientemente grande, y luego determina el entero M' que minimice la distancia entre $M'/(2^m)$ y el promedio obtenido.

El número de mensajes que necesita considerar B se puede acotar con la desigualdad de Chebyshev. Para esto, se definen variables aleatorias $X_1, \dots, X_t$ correspondientes a los mensajes, y $\bar{X}_t = (X_1 + \dots + X_t)/t$.

Luego, se tiene que $\mathbb{E}[\bar{X}_t] = p$, y por lo tanto se necesita que $\Pr[|\bar{X}_t - \mathbb{E}[\bar{X}_t]| \leq \frac{1}{2 \cdot 2^m}] < \varepsilon$. De la desigualdad de Chebyshev, se tiene que

$$\Pr\left[|\bar{X}_t - \mathbb{E}[\bar{X}_t]| \leq \frac{1}{2 \cdot 2^m}\right] \leq \text{Var}(\bar{X}_t) \cdot (2 \cdot 2^m)^2,$$

y también, usando que los mensajes son iid:

$$\begin{aligned} \text{Var}(\bar{X}_t) &= \left(\frac{1}{t}\right)^2 (\text{Var}(X_1) + \dots + \text{Var}(X_t)) \\ &= \left(\frac{1}{t}\right)^2 (tp(1-p)) \\ &= \frac{p(1-p)}{t} \end{aligned}$$

Luego, es suficiente que t cumpla

$$\frac{p(1-p)}{t} (2 \cdot 2^m)^2 \leq \frac{(2 \cdot 2^m)^2}{4t} < \varepsilon,$$

que sucede con $t > 2^m/\varepsilon$.

Problema 3

Se tiene un conjunto S de n elementos. Además, para cada l entre 1 y k , se tiene dado un conjunto $S_l \subseteq S$, tales que si $i \neq j$, entonces $S_i \neq S_j$, y además $|S_i| = |S_j| = r$.

Dado un conjunto de esa forma, donde $k < 2^{r-2}$, se necesita asignar uno de dos colores a cada elemento, de forma que cada S_i tenga al menos un elemento de cada color.

Describe y analice un algoritmo que permita encontrar una coloración como la descrita.

Solución Asignar colores de forma aleatoria. Luego, se acota la probabilidad de que esta sea una coloración. Para cada subconjunto i , se definen los eventos C_i , que representa que todos los elementos tengan color 0, y U_i , que indica que todos los elementos tienen color 1. Luego, se tiene que la probabilidad de que no sea una coloración es:

$$\Pr[\cup_{i=1}^k (C_i \cup U_i)] \leq \sum_{i=1}^k ((1/2)^r + (1/2)^r) = k \cdot (1/2)^{r-1}$$

luego, como $k < 2^{r-2}$, entonces $\Pr[\cup_{i=1}^k (C_i \cup U_i)] < 1/2$.

Problema 4

Se tiene como input un árbol binario completo con profundidad $2n$. Las hojas del árbol tienen valores de $\{0, 1\}$, y los nodos internos tienen conectivos $\wedge, \vee$, que se alternan de acuerdo a la profundidad, es decir, los hijos de un nodo $\wedge$ son nodos $\vee$, y los hijos de un nodo $\vee$ son nodos $\wedge$.

El costo de evaluar un árbol de este tipo se define como cantidad de hojas revisadas.

Parte 1 Proponga y analice un algoritmo aleatorizado, que en promedio revise 3^t hojas, y que siempre entregue un resultado correcto.

Idea de solución Si un nodo $\wedge$ tiene un hijo que evalúa a 0, entonces no es necesario evaluar el otro subárbol. Similarmente, si un nodo $\vee$ tiene un hijo 1, entonces se puede omitir el otro. Dado lo anterior, el algoritmo se basa en elegir aleatoriamente de forma uniforme uno de los hijos, y evaluar recursivamente ese árbol. Si el resultado permite determinar el valor del nodo, entonces no es necesario revisar el otro subárbol.

La cantidad de hojas revisadas en promedio se puede demostrar por inducción. En el caso base, cuando el árbol tiene altura 2, se deben considerar 4 casos: Si la raíz es $\wedge$ o $\vee$, y si evalúa 0 o 1:

- Raíz $\wedge$ que evalúa 1. En este caso, ambos hijos evalúan 1, y se deben revisar ambos. Como los conectivos se alternan en profundidad, entonces los hijos son nodos $\vee$ que evalúan a 1. En cada uno de ellos, la cantidad esperada de hojas revisadas depende del caso. Si ambas hojas son 1, entonces se revisa una hoja. Si una es 1, y la otra es 0, entonces en promedio se revisan $2 \cdot (1/2) + 1 \cdot (1/2) = 3/2$. En ambos casos, la cantidad esperada es a lo más $3/2$, por lo que en el árbol completo se revisan 3 hojas en promedio.
- Raíz $\vee$ que evalúa 1. Hay dos posibilidades, ambos hijos evalúan 1, o uno evalúa 0 y el otro 1. Si los dos son uno, entonces solo se revisa el primero, y por lo tanto la cantidad de hojas revisadas es a lo más 2. En el otro caso, un subárbol evalúa 1, y el otro 0. En el que evalúa 1, como es un nodo $\wedge$, solo se revisa una hoja. En el otro, ambos hijos son 0, o uno es 0 y el otro es 1. Si ambos son 0, solo se revisa 1, y si uno es 0 y el otro 1, se revisan $3/2$. Luego, en el subárbol que evalúa 0 se revisan a lo más $3/2$ hojas, y en el que evalúa 1, se revisa 1 hoja, por lo tanto, la cantidad esperada de hojas revisadas en el árbol completo es: $(1/2) \cdot 1 + (1/2) \cdot (5/2) = 3$.

Los otros dos casos son análogos. La demostración termina por inducción en la altura. Para esto se consideran los dos primeros niveles del árbol. Por el mismo argumento de antes, se revisan 3 de los cuatro subárboles, y en cada uno, por hipótesis de inducción, se revisan 3^k hojas. Luego, si el árbol tiene altura $2(k+1)$, se revisan 3^{k+1} hojas en promedio.

Parte 2 Modifique el algoritmo anterior para que siempre revise a lo más una fracción fija de la cantidad de hojas, y determine la probabilidad de que entregue un resultado.

Idea de solución El mismo algoritmo, pero con un contador de hojas revisadas. Si la cantidad supera la fracción dada, entonces termina sin resultado, y en caso contrario entrega el resultado correcto.

Para acotar la probabilidad de termino, se considera una variable aleatoria X , que indica la cantidad de hojas revisadas por el algoritmo de la parte 1. Luego, dada una fracción fija p , la probabilidad de falla es $\Pr[X > p(4^k)]$, que es menor o igual a $3^k/(p4^k)$ por la desigualdad de Markov.