Universidad del Valle de Guatemala
Análisis y Diseño de Algoritmos - Sección 10
María Fernanda Estrada 14198
13/05/2020

Tarea 6
Análisis competitivo y algoritmos on-line

1. **El año es 2012. Mabel, quien acaba de mudarse al norte de Canadá, irá a esquiar mañana por primera vez en su vida. Ella puede comprar sus ski's de una vez por B dólares o rentar ski's a un dólar por día. Mabel no sabe cuántas veces irá a esquiar en el futuro, pero quisiera gastar lo menos posible.**
   a. **Si Mabel tuviera el dinero necesario podría ir con Walter Mercado, de quien recibiría la cantidad n de días que ella esquiará en toda su vida (aparte de mucho, pero mucho amor). ¿Cuánto, en este caso, gastaría Mabel en ski's?**
   Gastaría $min\,(n,\ B)$, es decir el mínimo entre la cantidad n de días de alquiler y los B dólares que cuestan los ski's.
   b. **Como a Mabel no le alcanza para ir con Walter Mercado piensa que será mejor probar alquilando cada día, y si llegara a esquiar por B días se verá convencida de comprar sus ski's el B-ésimo día, evitándose ese alquiler y cualquier alquiler futuro. ¿Cuánto estimaría gastar en el peor de los casos, siguiendo este plan? Determine la proporción entre este gasto y el del peor de los casos del inciso anterior.**
   Como se menciona, el peor de los casos es que alquile hasta el B-ésimo día y que después igualmente gaste en comprar los ski's. Esto causa un costo total de $(B-1)+B=2B-1$. En el inciso anterior el peor de los casos es que Mabel gaste únicamente B dólares, por lo que la proporción es $\frac{2B-1}{B}=2-\frac{1}{B}$.
   c. **Mabel piensa que su plan posiblemente no sea el mejor, y que quizá podría reducir la proporción calculada. Desesperada, Mabel invierte su dinero en consultar por teléfono a Urbano Madel quien, por un precio más accesible, le puede resolver su inquietud. La respuesta de Urbano Madel es que no, la proporción entre esos gastos no puede ser reducida. Demuestre que Urbano Madel tiene razón.**
   Si tomamos que el gasto total es de $(j-1)+B$, siendo j la cantidad de alquiler, podemos tomar dos escenarios. El primero es cuando la cantidad de días en alquiler j es menor o igual que B y el otro es cuando j es mayor a B. Esto nos da la proporción para ambos casos:
   $$\frac{j-1+B}{min(j,\ B)}=\frac{j-1+B}{j}=1+\frac{B-1}{j} \quad \text{\# si j} \leq \text{B}$$
   $$\frac{j-1+B}{min(j,\ B)}=\frac{j-1+B}{B}=1+\frac{j-1}{B} \quad \text{\# si j} > \text{B}$$
   Para determinar si estos resultados son mejores que el inciso anterior, reemplazamos por B en ambos casos para encontrar lo mínimo que podrían dar. Al evaluar, fueron los mismos resultados y es igual a la proporción del inciso b.

$$1 + \frac{(B-1)}{B} = 2 - \frac{1}{B}$$

2. **Suponga que Mabel dice "Qué me importa. Voy a probar esquiar alquilando por k días y si no me he aburrido me compro mis esquís". k es una cantidad de días entre 0 y B−1. La probabilidad de que se elija una cantidad de días k es igual a $\alpha\varrho^k$, donde α = $\frac{-1}{^{B}-1}$ y ϱ = $\frac{}{-1}$. Alguien con clarividencia podría decirle a Mabel cuántos días esquiará en toda su vida. Esa cantidad, n, podría ser mayor a B. Como Mabel elegirá comprar sus ski's al cabo de 0 ≤ k ≤ B−1, nos interesa el caso en que $n \leq B$, expresándose con $n = B$ que Mabel esquiará por suficiente tiempo como para alquilar por B−1 días y aun así comprar sus ski's.**

   a. **Provea $E[X]$, el costo esperado en el que incurrirá Mabel eligiendo alquilar por k días. El valor de X depende de k y tiene la distribución de probabilidad especificada previamente. Recuerde que el rango posible de días está entre 0 y B−1.**

   Usamos fórmula de valor esperado

   $$E[X] \ = \Sigma_{k=0}^{B-1} \alpha\rho^k (costo\ k - \acute{e}simo\ d\acute{\imath}a)$$

   El costo sería la suma de los días en los que podría comprar los ski's (n - 1) y los días en los que ya no los comprará (después de n días). La ecuación entonces es:

   $$E[X] \ = \Sigma_{k=0}^{n-1} \alpha\rho^k (costo\ antes\ de\ n-1) \ + \Sigma_{k=0}^{B-1} \alpha\rho^k (costo\ despu\acute{e}s\ de\ n)$$

   Si k se encuentra dentro de los primeros n - 1 días, se agregará el costo B de haber comprado los ski's. Si cae después de los n días, este costo no se suma porque ya no los comprará. Nuestro valor esperado es:

   $$E[X] \ = \Sigma_{k=0}^{n-1} \alpha\rho^k (k+B) \ + \Sigma_{k=0}^{B-1} \alpha\rho^k (n)$$

   b. **Demuestre que este algoritmo usado por Mabel es c-competitivo y encuentre el valor de c.**

   Si seguimos trabajando las ecuaciones del valor esperado, obtenemos el siguiente resultado:

   $$E[X] \ = \alpha\Sigma_{k=0}^{n-1} \rho^k k + \alpha B \Sigma_{k=0}^{n-1} \rho^k + an\Sigma_{k=0}^{B-1} \rho^k$$

   Por recomendación, se ingresaron a Wolfram y continúa el desarrollo

   $$E[X] = \frac{\alpha(n-1)\rho^{n+1} - n\rho^n + \rho}{(\rho-1)^2} + \frac{\alpha B(\rho^n - 1)}{\rho-1} + \frac{\alpha n(\rho^B - \rho^n)}{\rho-1}$$

   Si $B = \frac{\rho}{\rho-1}$, y si multiplicamos los dos últimos factores por $\frac{\rho-1}{\rho-1}$ para poder factorizar, obtenemos:

   $$E[X] = \frac{\alpha}{(\rho-1)^2}(((n-1)\rho^{n+1} - n\rho^n + \rho) + \rho(\rho^n - 1) + n(\rho - 1)(\rho^B - \rho^n))$$

   $$E[X] = \frac{\alpha}{(\rho-1)^2}(n\rho^{n+1} - \rho^{n+1} - n\rho^n + \rho + \rho^{n+1} - \rho + n\rho^{B+1} - n\rho^{n+1} - n\rho^B + n\rho^n)$$

   $$E[X] = \frac{\alpha}{(\rho-1)^2}(n\rho^{B+1} - n\rho^B) = \frac{\alpha}{(\rho-1)^2} n\rho^B(\rho - 1) = \frac{\alpha}{\rho-1} n\rho^B$$

   Si reemplazamos α, tenemos finalmente

   $$E[X] = \frac{\rho^B}{\rho^B - 1} n = \frac{\rho^B}{\rho^B - 1} opt$$

   Como sabemos que n es lo que el algoritmo óptimo gastaría, tenemos que el valor de c viene dado por

   $$c = \frac{\rho^B}{\rho^B - 1}$$

3. **Conseguimos trabajo como patoj@ chispud@ y nuestra tarea es colocar elementos con peso entre 0 y 1 kilogramos en cajas cuya capacidad máxima es de 1 kilogramo. A lo largo de un día de trabajo van entrando elementos, pero cada elemento se nos entrega hasta después de que hayamos asignado el anterior a una caja. No se nos permite reorganizar elementos ya asignados. Al final del día nos dan un bono por el espacio que hayamos logrado ahorrar.**
La empresa nos ha provisto de una cantidad de cajas infinita. Demuestre que seguir el algoritmo de guardar el elemento entrante en la primera caja en la que quepa (o una nueva, si no cabe en ninguna de las ya ocupadas) es 2-competitivo.
Primero debemos encontrar una cota inferior a la cantidad de cajas usadas por el algoritmo. Si tenemos n elementos, el peso total viene dado por $x = \Sigma_{i=1}^{n} p_i$. Nuestro algoritmo usa c cajas, entonces al menos c - 1 almacenan igual o más de 0.5 kilos. Como las c - 1 cajas deben estar llenas al menos a la mitad, el peso total debe ser mayor a c - 1 cajas de 1 kilo llenas a la mitad. Es decir
$$\Sigma_{i=1}^{n} p_i > \frac{c-1}{2}$$
Ahora despejamos para c - 1
$$2\Sigma_{i=1}^{n} p_i > c - 1$$
Por último, como el peso total de los elementos es cota inferior, comprobamos que seguir el algoritmo es 2-competitivo:
$$2opt \geq 2\Sigma_{i=1}^{n} p_i > c - 1$$
$$2opt + 1 > c$$

4. **El paging problem nos enfrenta a un sistema de memoria separado en dos niveles que almacenan páginas de memoria. Un nivel es la caché y el otro es lo que llamaremos "memoria lenta". Cuando se solicita una página de memoria, si la página está en cache se considera insignificante el costo de acceso. Si la página no está en caché debemos primero acceder a ella en la memoria lenta y luego moverla a cache (efectivamente reemplazando a otra en caché si ésta está llena) para presentarla al solicitante. La capacidad de la memoria lenta es de N páginas, donde N es mucho mayor a la capacidad k de cache. Además, la caché no puede contener páginas que no están en la memoria lenta. Suponemos que el costo de acceder a una página en memoria lenta y trasladarla a cache es 1.**
    a. **Demuestre que tratar caché como una pila (remover de caché la página que haya sido copiada desde memoria lenta más recientemente) no es competitivo.**
    Tenemos una pila caché con páginas de la 1 a la k, y una extra x al final. Tenemos una secuencia de solicitudes a las páginas de la 1 a ka k-1, y luego n solicitudes alternadas de las páginas k y x. Las solicitudes de las páginas k y x provocan un reemplazo siempre porque tratamos a la caché como una pila, por lo que causa un costo de n. Ahora, un algoritmo óptimo intentaría que las páginas k y x se quedarán en caché, ahorrando los gastos de solicitud. Esto causa que el costo del algoritmo óptimo sea de 1. Se puede observar que el primer algoritmo es costo n y el optimizado es 1; no hay

constantes que permitan afirmar que $LIFO \leq c * opt + \alpha$, por lo que no hay competitividad.

b. **Demuestre que, ante una secuencia arbitraria de páginas solicitadas, aplicar la heurística *least recently used* (abreviado LRU) de reemplazar la página en caché "más vieja" (es decir, aquella cuya última solicitud se haya hecho antes que las de todas las demás) es *k*-competitivo (donde k es la capacidad de la caché). Para resolver este ejercicio complete los siguientes incisos:**

   i. **Explique cómo se comportará el adversario malicioso.**
   Buscará solicitudes fuera de la caché del algoritmo que usa LRU pero que sí estén en su caché, por lo que haría al algoritmo gastar en cada solicitud y el adversario no.

   ii. **Demuestre que el adversario malicioso debe incurrir en gastos en algún momento.**
   El adversario solicitará de primero una página a. Luego, después de solicitar las otras k - 1 páginas, la primera página a será el próximo reemplazo. Para hacernos reemplazar a la página a, el algoritmo óptimo tendrá que solicitar una página que esté fuera de su propia caché, causando un costo de 1.

   iii. **Demuestre que la proporción (cociente) entre los gastos incurridos por LRU y el óptimo es *k*.**
   Como se mencionó en el inciso anterior, el algoritmo óptimo tiene un gasto de 1. Por otro lado, el algoritmo LRU tiene un gasto de k porque el adversario nos solicitó k páginas fuera de nuestra caché. La proporción entonces es $\frac{k}{1} = k$.