Universidad del Valle de Guatemala Christopher Sandoval 13660
Análisis y Diseño de Algoritmos - Sección 10 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?**
      No sabemos por quién decidirá, pero sabemos que quiere gastar lo menos posible. Así que Mabel gastará el mínimo entre n y B, ya sea si el mínimo es alquilar o comprar 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.**
      Si Mabel sigue este plan, gastará B - 1 días en alquiler y luego B dólares en comprar los ski's después. Al sumar este costo total, tenemos que es de B - 1 + B = 2B - 1, en el peor de los casos. La proporción viene dada por (2B - 1)/B = 2 - 1/B. Esto porque el peor de los casos del inciso anterior es que Mabel compre los ski's por B dólares.
   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.**
      Usaremos la ecuación de proporción de inciso anterior. Sin embargo, como no podemos asegurar que ocurrirá el peor de los casos, supondremos que alquilará una cantidad j los ski's. Esta j puede ser menor o igual a B o mayor. Si j ≤ B, se tiene una proporción de $\frac{j-1+B}{min(j,B)} = 1 + \frac{B-1}{j}$. Si j > B, la proporción sería de $\frac{j-1+B}{min(j,B)} = 1 + \frac{j-1}{B}$. Como deseamos saber si estas dos proporciones son mejores a la del inciso anterior, tenemos que determinar los mínimos valores en las que pueden resultar. En ambos casos deberíamos reemplazar j por B, ya que eso nos daría el mínimo valor posible. Al reemplazar obtenemos la proporción $1 + \frac{(B-1)}{B} = 2 - \frac{1}{B}$. Como ambos son iguales después de reemplazar por B,

notamos que también esta proporción es igual a la del inciso anterior; demostramos que Urbano Madel tiene razón.

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 $\alpha\rho^k$, donde $\alpha = \frac{-1}{B-1}$ y $\rho = \frac{}{-1}$. 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 $0 \leq k \leq B-1$, 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 $B-1$.**

   Fórmula costo esperado, considerando como variante en qué día comprará los ski's

$$E[X] = \sum_{k=0}^{B-1} \alpha\rho^k(costo\ k\ día)$$

   El costo cambia si compra los ski's antes de n-1 días o si se pasan los n días y ya no los compra.

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

   Si decide comprar antes de los n - 1 días, se agrega el costo de alquiler y el costo de los ski's. Si se pasan los n días, solo será el costo de los n días.

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

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

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

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

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

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

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

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

   Si n es lo que consume el algoritmo optimizado y considerando que ρ y B son constantes, determinamos que el valor de c es

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

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.

   Si tenemos un set de elementos, su peso total vendría siendo la suma de ellos $x = \sum_{i=1}^{n} b_i$.

   Esta ecuación sirve como cota inferior a la cantidad de cajas usadas; si cada caja contiene máx 1 kilo y tenemos x kilos, necesitaríamos x cajas. A partir de esto, suponemos que el algoritmo usa k cajas, en donde k - 1 cajas almacenarán un peso mayor o igual a medio kilo. Esta suposición es válida porque si la caja tiene un peso menor a medio kilo, cuando entre otro elemento y se almacene en la misma caja, ésta superará el medio kilo. Como las k - 1 cajas deben estar llenas al menos a la mitad, el peso total x debe ser mayor a k - 1 cajas de 1 kilo llenas a la mitad. Esto se traduce a $\sum_{i=1}^{n} b_i > \frac{k-1}{2}$. Despejamos para k-1 luego. Al principio determinamos que el peso total es cota inferior para las k cajas usadas, por lo que $2opt \geq 2\sum_{i=1}^{n} b_i > (k-1)$. Por transitividad, $m < 2opt + 1$ y esto ya demuestra que el algoritmo es 2-competitivo.

4. **El paging problem nos enfrenta a un sistema de memoria separado en dos niveles que almacenan páginas de memoria. Un nivel es la cache 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 cache debemos primero acceder a ella en la memoria lenta y luego moverla a cache (efectivamente reemplazando a otra en cache 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 cache 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 cache como una pila (remover de cache la página que haya sido copiada desde memoria lenta más recientemente) no es competitivo.**

      No existe competitividad por la diferencia de costos entre un algoritmo y un algoritmo óptimo. El primer algoritmo (LIFO) tiene costo n porque si se hacen n solicitudes a dos páginas x y y de forma alternada, la última página que entró se reemplaza cuando se solicita una página que no está en caché. El algoritmo óptimo intenta que las dos páginas x y y se queden en caché para evitar el reemplazo, por lo que su costo es de 1. No hay constantes que al multiplicar y

sumar al algoritmo óptimo puedan llegar a ser mayor que el algoritmo LIFO, porque podemos hacer a n muy grande.

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. Recuerde que el adversario maneja su propia memoria lenta y su propia caché.**

      El adversario intentaría hacer solicitudes a su propia caché, pero se asegura que estas páginas estén fuera de la caché del algoritmo LRU. Esto hace que el adversario no tenga costos, pero el algoritmo gastaría con cada solicitud.

   ii. **Demuestre que el adversario malicioso debe incurrir en gastos en algún momento.**

      La primera página que se nos solicita es m. Eventualmente el adversario realizará k - 1 solicitudes de páginas distintas y la página m deberá ser reemplazada. Hasta el momento, el adversario solo ha hecho solicitudes de páginas que están en su caché, por lo que ya agotó el tamaño de su caché con k solicitudes. Si quiere reemplazar a la página inicial m, deberá hacer forzosamente una solicitud fuera de su caché, causando un costo de 1.

   iii. **Demuestre que la proporción (cociente) entre los gastos incurridos por LRU y el óptimo es k.**

      Sabemos que el adversario ha hecho k solicitudes fuera de la caché del LRU, por lo que el gasto que ha hecho el algoritmo LRU es de k. Por otro lado, se demostró en el inciso anterior que el único gasto que tendría el algoritmo óptimo es de 1. Entonces, la proporción entre los gastos incurridos es k/1= k.