Pauta de Corrección

Segundo Certamen
Algoritmos y Complejidad

22 de diciembre de 2018

1. Para simplificar la discusión, numeraremos las estaciones de recarga en orden de distancia creciente del origen.

   El planteado claramente es un algoritmo voraz. Para demostrar que da un óptimo, vemos las tres propiedades:

   Greedy choice: Debemos demostrar que recargar combustible lo más tarde posible como la primera recarga es parte de una secuencia óptima. Sea esta recarga en la estación i.

   Consideremos una secuencia óptima. Si incluye recargar en la estación i, estamos listos. Si no, recarga en una estación $j < i$ (ya que i es la última alcanzable), podemos postergar la recarga de combustible a la i y continuar con la secuencia original. Esto da el mismo número de recargas, hay una secuencia óptima que incluye la elección voraz.

   Inductive substructure: El problema menor que resulta al elegir la estación i es alcanzar el objetivo con el mínimo de recargas desde s_i, considerando solo las estaciones $\{i+1,\ldots,n\}$. No hay restricciones externas, toda solución a este nuevo problema puede combinarse con recargar en la estación i.

   Optimal substructure: La solución óptima a partir de s_i y recargar en las estaciones $\{i+1,\ldots,n\}$ junto con recargar en i da la solución óptima (mínimo número de recargas).

   Como se cumplen los tres requisitos, da un óptimo.

Puntajes

Total	25
Greedy choice	10
Inductive structure	7
Optimal substructure	8

2. Llamemos $A(n)$, $B(n)$ y $C(n)$ los tiempos de ejecución respectivos. Del problema extraemos las recurrencias. En la notación del problema maestro al reverso del enunciado:

Algoritmo A: La recurrencia da $a = 3$, $b = 3$, $f(n) = Theta(n \log n) + Theta(n) = Theta(n \log n)$. Tenemos $\alpha = \log_3 3 = 1$, por el cuarto caso del teorema maestro (tenemos $\beta = 1$):

$$\begin{aligned} A(n) &= \Theta(n^\alpha \log^{\beta+1} n) \\ &= \Theta(n \log^2 n) \end{aligned}$$

Algoritmo B: Resulta la recurrencia:

$$B(n+2) = B(n+1) + B(n) + c$$

Es claro que $B(n) \geq F_n$, un número de Fibonacci. Esto crece exponencialmente.
En detalle, usando la función generatriz:

$$g(z) = \sum_{n \geq 0} B(n) z^n$$

aplicando propiedades a la recurrencia obtenemos:

$$\frac{g(z) - B(0) - B(1)z}{z^2} = \frac{g(z) - B(0)}{z} + g(z) + \frac{c}{1-z}$$

En fracciones parciales:

$$g(z) = \frac{(B(1) - B(0))z}{1 - z - z^2} + \frac{c + B(0)}{1 - z - z^2} - \frac{c}{1-z}$$

Del torpedo vemos:

$$B(n) = (B(1) - B(0)F_n + (c + B(0))F_{n+1} + c$$

Algoritmo C: Para el teorema maestro, $a = 7$, $b = 4$, $f(n) = Theta(n^2)$. El valor crítico es $\alpha = \log_4 7 < 2$ (ya que $4^2 = 16$), estamos en el quinto caso con $c = 2$ ya que:

$$\begin{aligned} 7\left(\frac{n}{4}\right)^2 &= \frac{7n^2}{16} \\ &= \frac{7}{16} n^2 \end{aligned}$$

que permite elegir k en el rango $7/16 < k < 1$. O sea:

$$C(n) = \Theta(n^2)$$

Para n grande, el mejor es el algoritmo A.

Puntajes

Total	30
Algoritmo A	9
Algoritmo B	9
Algoritmo C	9
Conclusión	8

3. Ordenemos las ciudades por las posiciones del par a lo largo de la ribera izquierda. Debemos considerar la opción de agregar el puente $a_i - b_i$. Si no lo agregamos, el máximo hasta acá es el que resulta de considerar solo los puentes $1, \ldots, i-1$; si lo agregamos es el máximo considerando los puentes $1, \ldots, j$ (acá j es el último puente con el que el puente i no interfiere, o sea el máximo j para el que $b_j < a_i$) más 1. Esto da la recurrencia:

   $$m_i = \text{máx}\{m_{i-i}, m_j + 1\}$$

   condición de contorno es $m_0 = 0$, nos interesa m_n. Almacenamos valores intermedios en un arreglo de una dimensión, que vamos llenando desde el inicio, buscando para cada i el j relevante.

Puntajes

Total	30
Planteo de la recurrencia	15
Condición inicial	10
Orden de llenado	5

4. Veamos los eventos relevantes en cada caso, considerando independientes los lanzamientos.

   Alice: Dos lanzamientos, eventos son cómo se interpretan en el rango 1 a 3:

1	HH	p^2
2	HT	$p(1-p)$
3	TH	$p(1-p)$
–	TT	$(1-p)^2$

   Vemos que solo si $p = 1/2$ las tres primeras tienen la misma probabilidad.

   Bob: Los resultados de los lazamientos son los mismos que para Alice, pero nos interesa el número de H ahora:

1	TT	$(1-p)^2$
2	HT, TH	$2p(1-p)$
3	HH	p^2

   Vemos que solo si $p = 1/2$ son iguales las probabilidades de 1 y 3, pero en tal caso la probabilidad de 2 es el doble de las anteriores. No funciona para ningún valor de p.

   Charlie: Nuevamente listamos los resultados posibles de cada experimento y sus interpretaciones:

1	HTT	$p(1-p)^2$
2	THT	$p(1-p)^2$
3	TTH	$p(1-p)^2$
–	$\cdots$	$1-3p(1-p)^2$

   Esta propuesta da probabilidades iguales, como solicitado, para todo p.

   Incidentalmente, el número esperado de intentos hasta tener éxito si la probabilidad de éxito es p (y falla $q = 1-p$) es:

$$\begin{aligned}\sum_{k\geq 0} kq^{k-1}p &= \frac{p}{q}\sum_{k\geq 0} kq^k \\ &= \frac{p}{q}\cdot\frac{q}{(1-q)^2} \\ &= \frac{1}{p}\end{aligned}$$

   Con las probabilidades de éxito calculadas antes, tenemos que Alice en promedio lanza la moneda un número de veces dado por:

$$2\cdot\frac{1}{p^2+2p(1-p)} = \frac{2}{p(2p-1)}$$

   mientras el número de lanzamientos promedio de Charlie es:

$$3\cdot\frac{1}{3p(1-p)^2} = \frac{1}{p(1-p)^2}$$

Puntajes

Total		30
Análisis de la propuesta de Alice	8	
Análisis de la propuesta de Bob	8	
Análisis de la propuesta de Charlie	8	
Conclusión	6	

5. Suponemos costo unitario para insertar o eliminar un elemento de la lista. Tiene sentido ahorrar una unidad con cada elemento en la estructura, para pagar su eliminación posterior. Definimos entonces $\Phi(F) = |F|$. Inicialmente, la fila está vacía, $\Phi(F_i) = 0$, y siempre es $\Phi(F) \geq 0$. Para la operación orderedpush(x), si el largo al comienzo de la operación era a y termina en b (necesariamente $a \geq b$; quitamos $a-b+1$ elementos, con costo $a-b+1$; agregamos 1 elemento, con costo 1), el costo amortizado es:

   $$a - b + 1 + 1 + \Phi(b) - \Phi(a) = 2$$

   El costo amortizado de la operación pop() es igual que arriba, pero con $a = b+1$. En resumen, el costo amortizado de cada operación es $O(1)$.

Puntajes

Total	30
Función potencial	15
Condición inicial y final	5
Operación orderedpush(x)	4
Operación pop	4
Conclusión	2

HvB/LaTeX2ε