Nombre: Javier Ignacio Gómez Peña
19.516.104-6

Pregunta 1

a) Sabemos que un valor n se puede formar con NlogDolares si:
$n = a \cdot 3 + b \cdot 5$, con $a, b \in \mathbb{N}$
Si comenzamos a ver qué valores podemos formar con NlogDolares tenemos los siguientes: {3, 5, 6, 8, 9, 10, 11, 12, 13…}
Luego, podemos notar que el 8, 9 y 10 son tres valores consecutivos que se pueden formar. Cómo tenemos un billete de NlogDolares que equivale a 3, siempre podremos sumarlo cuantas veces queramos a estos tres valores, por lo que a partir de estos obtenemos el { (8+3), (9+3), (10+3) } que corresponden al {11, 12, 13}, por lo que, a partir de aquí, siempre podremos formar cualquier valor.
Por lo tanto, el valor más grande que no se puede formar con NlogDolares es el 7.

b) Queremos demostrar que $\forall n > 7 \Rightarrow n = a \cdot 3 + b \cdot 5$, con $a, b \in \mathbb{N}$
Usando inducción fuerte:
<u>Casos base:</u>
$n_0 = 3 \cdot 1 + 5 \cdot 1 = 8$
$n_1 = 3 \cdot 3 + 5 \cdot 0 = 9$
$n_2 = 3 \cdot 0 + 5 \cdot 3 = 10$

<u>Caso inductivo:</u>
Queremos demostrar que $\forall n > 7$ se puede escribir de la forma 3a + 5b. Tenemos los casos base para n=8, n=9 y n=10, por lo que nuestra demostración se reduce a demostrar que $\forall n > 10$ se puede escribir de la forma 3a + 5b. Tomemos j arbitrario. Planteamos nuestra H.I. tal que j cumple con lo pedido (se puede escribir como 3a + 5b). Si tomamos $j = n-3 \Rightarrow 8 \leq n-3 < n$. Por H.I. fuerte tenemos que n-3 también cumple con lo pedido. Por lo tanto, n-3 = 3a + 5b. Para obtener n basta sumar 3, es decir $n = 3(a+1) + 5b$, con $(a+1), b \in \mathbb{N}$.
Por lo tanto, n se puede escribir como 3a + 5b, lo que demuestra lo pedido.

c) Queremos encontrar todas las secuencias (o posibles combinaciones) de NlogDolares tal que valgan exactamente un valor d. Sabemos que está prohibido vender a un menor precio que n = 7 (incluyéndolo), por lo que $d \geq 8$ necesariamente.
Para encontrar las secuencias distintas que valen d, usaremos inducción fuerte.
Primero, encontraremos un conjunto A_d de las posibles secuencias que valen d, para luego contarlas y así obtener el número total de ellas.
Para esto, definiremos dos operadores que serán necesarios para plantear esto:
- ♠: Concatenación de un elemento m por la izquierda de una secuencia de NlogDolares.
  Ej: {(3, 3, 3)} ♠ 5 = {(3, 3, 3, 5)}

- ♣: Aplica ♠ a cada una de las secuencias de un conjunto de secuencias.
  Ej: A = {(3, 3, 3), (5, 5, 5)}
  A ♣ 5 = {(3, 3, 3, 5), (5, 5, 5, 5)}

Plantearemos los casos menores a 8, y los casos d = 9, d = 10 y d = 11 (esto ya que sabemos que a partir de estos valores podemos formar los valores mayores) como casos base para luego poder definir el caso para d como un caso inductivo.

Casos base:

- $d = 3 \rightarrow A_3 = \{3\}$
- $d = 5 \rightarrow A_5 = \{5\}$
- $d = 6 \rightarrow A_6 = \{(3, 3)\}$
- $d = 8 \rightarrow A_8 = \{(3, 5), (5, 3)\}$
- $d = 9 \rightarrow A_9 = \{(3, 3, 3)\}$
- $d = 10 \rightarrow A_{10} = \{(5, 5)\}$
- $d = 11 \rightarrow A_{11} = \{(3, 5, 3), (5, 3, 3), (3, 3, 5)\}$

Caso inductivo:

- $A_d = (A_{d-3} \clubsuit 3) \cup (A_{d-5} \clubsuit 5)$

Así, por ejemplo, para d = 14:

$A_{14} = (A_{11} \clubsuit 3) \cup (A_9 \clubsuit 5)$
$A_{14} = \{(3, 5, 3, 3), (5, 3, 3, 3), (3, 3, 5, 3)\} \cup \{(3, 3, 3, 5)\}$
$A_{14} = \{(3, 5, 3, 3), (5, 3, 3, 3), (3, 3, 5, 3), (3, 3, 3, 5)\}$

Luego basta tomar el número de elementos del conjunto A_d para tener el número de secuencias distintas que valen d.

d) Código se incluye adjunto vía U-Cursos

Pregunta 2

a) Sea Q_n la mínima cantidad de movimientos para mover una torre de n pisos desde A hasta B y R_n la cantidad de movimientos que se necesitan para retornar la misma torre desde B hasta A.

Queremos mostrar lo siguiente:

$$Q_n = \begin{cases} 0 & \text{si } n = 0 \\ 2R_{n-1} + 1 & \text{si } n > 0 \end{cases}$$

$$R_n = \begin{cases} 0 & \text{si } n = 0 \\ Q_n + Q_{n-1} + 1 & \text{si } n > 0 \end{cases}$$

Veamos primero Q_n :

Se quiere mover n discos de A hasta B. Podemos mover n-1 discos de A hasta C (que sería equivalente a mover los n-1 discos de B hacia A, en números de movimientos). Esto toma R_{n-1} movimientos ya que equivale a mover discos de B hacia A (en número de movimientos).

Luego, movemos el disco restante de A hacia B (toma apenas 1 movimiento).
Ya que tenemos el disco más grande en B, falta mover los restantes de C a B (que quedarían en el orden correcto al moverlos). Mover los discos de C a B toma R_{n-1} movimientos ya que es equivalente a mover n-1 discos de B hacia A.
Si sumamos todos los movimientos:

- Mover n-1 discos de A hacia C = R_{n-1}
- Mover 1 disco de A hacia B = 1
- Mover n-1 discos de C hacia B = R_{n-1}
- $\rightarrow Q = R_{n-1} + R_{n-1} + 1 = 2R_{n-1} + 1$

Veamos R_n :

Inicialmente tenemos n discos en B y movemos n-1 discos hacia A. Esto toma R_{n-1} movimientos. Luego movemos el disco restante hacia C (esto toma sólo 1 movimiento). Luego movemos los n-1 discos de A hacia B, lo que toma Q_{n-1} movimientos (por definición de Q_n) y movemos el disco de C hacia A, que toma de nuevo, un solo movimiento. Para terminar, movemos los n-1 discos de B hacia A (que toma también R_{n-1} movimientos). Después de esto, tenemos los n discos en A.
Si sumamos los movimientos:

- Mover n-1 discos de B hacia A = R_{n-1}
- Mover un disco de B hacia C = 1
- Mover n-1 discos de A hacia B = Q_{n-1}
- Mover un disco de C hacia A = 1
- Mover n-1 discos de B hacia A = R_{n-1}
- $\rightarrow R_n = R_{n-1} + 1 + Q_{n-1} + 1 + R_{n-1} = Q_{n-1} + 1 + 2R_{n-1} + 1$ y sabemos que $2R_{n-1} + 1 = Q_n$
- $\rightarrow R_n = Q_{n-1} + 1 + Q_n$
- $\rightarrow R_n = Q_n + Q_{n-1} + 1$

Con esto, tenemos lo que se pedía mostrar.

b) Código se incluye adjunto vía U-Cursos

Pregunta 3

a) Conocemos el algoritmo utilizado. Sabemos que la probabilidad de guardar el primer elemento (independiente de si lo descartamos después o no) es 1. También sabemos que la probabilidad de guardar un elemento i (y descartar el anterior) es de 1 / i.
Queremos demostrar que, para cualquier elemento de una lista de largo n, la probabilidad de escogerlo (guardarlo y luego conservarlo hasta el final) es de 1 / n.
Para cualquier elemento de una lista, independiente de que haya pasado con los elementos anteriores, la posibilidad de guardarlo es P(i) = (1 / i), esto es por lo indicado en el algoritmo. Luego, para conservarlo, es necesario **no guardar** los elementos restantes de la lista.
Definamos P'(i) = 1 – (1 / i) como la probabilidad de no elegir un elemento i de la lista.
Esto es equivalente a P'(i) = (i -1) / i

Luego, la probabilidad $P_c(i)$ de escoger (conservar) el elemento i de una lista de n elementos, será:

$$P_c(i) = P(i) \cdot P'(i+1) \cdot P'(i+2) \cdot \ldots \cdot P'(n-1) \cdot P'(n)$$

$$P_c(i) = (1 / i) \cdot (i / (i+1)) \cdot ((i+1) / (i+2)) \cdot \ldots \cdot ((n-2) / (n-1)) \cdot ((n-1) / n)$$

Simplificando nos queda:

$$P_c(i) = 1 / n$$

Quedando demostrado esto para todo elemento i en una lista de largo n.

b) Código se incluye adjunto vía U-Cursos