Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación
Universidad de Chile
CC3101-01 Matemáticas de Discoteca para la Computación

Profesor: Pablo Barceló
Auxiliares: Tomás Martínez, Ignacio Riego

Tarea 1: Inducción

P1. En el reino de Nlogonia se quiere eliminar todos los billetes exepto los de 3 y de 5 NlogDolares. Un grupo de mercaderes está nervioso de que exisitan valores que no se pueden formar solo con esos billetes. Decimos que un valor n se puede formar con esos billetes si existen a, b positivos tales que $n = 3a + 5b$

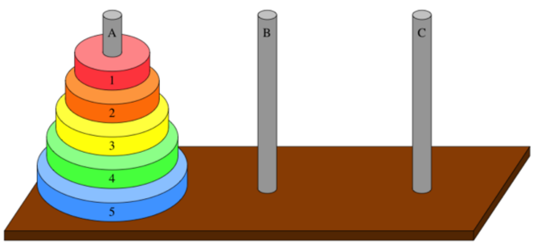
a) Para evitar este problema, el rey quiere prohibir vender productos a precios que no se puedan formar con esos billetes. Encuentre cual es el valor n más grande que no se puede formar con estos billetes para ayudar al rey a pasar su ley.

b) Los mercaderes siguen intranquilos. Exijen una demostración rigurosa de que ese es el n más pequeño. Pruebe usando inducción fuerte que todo número mayor que n puede formarse con esos billetes.

c) Una vez convencidos de que funciona, a los mercaderes de Nlogonia les encantó el nuevo sistema y se entretienen durante horas haciendo secuencias con sus nuevos billetes. Encuentre un método basado en la inducción fuerte para calcular cuantas secuencias distintas se pueden hacer que valgan exactamente d Nlogdólares. (Estas secuencias son sensibles al orden, por ejemplo "$3, 5$" es distinta de "$5, 3$")

f) Escriba un programa, basado en la inducción fuerte, que reciba un número d y entregue la cantidad de secuencias de billetes distintos que valen exactamente d o entregue 0 en caso de que no haya ninguna secuencia posible que valga en total d.

P2. Consideremos el famoso problema de las torres de Hanoi. Se trata de una serie de discos, todos distintos, puestos uno encima del otro, que van desde el más pequeño arriba hasta el más grande abajo. Tenemos 3 posiciones donde se pueden dejar los discos (A, B y C) y debemos mantener la condición de nunca poner un disco más grande sobre uno más pequeño.



a) En este problema vamos a agregar una nueva restricción, solamente podemos mover discos desde A hacia B, de B hacia C o de C hacia A. Ningun otro movimiento está permitido.

Con esta nueva restricción llamaremos Q_n a la mínima cantidad de movimientos para mover una torre de n pisos desde A hasta B y llamaremos R_n a la cantidad de movimientos que se necesitan para retornar la misma torre desde B hasta A. Muestre que se cumple lo siguiente:

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

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

b) Escriba un programa que para una torre de n pisos le dé instrucciones al usuario de cómo mover los discos para pasarla de posición A a posición B. (las instrucciones son de la forma $'A \to B'$ que significa tomar el disco más chico de A y ponerlo arriba de B, por supuesto todos los movimientos deben ser legales usando la restricción de la parte a).

P3 Un problema común para los algoritmos aleatorios es seleccionar entre una lista de n elementos seleccionar aleatoriamente alguno de ellos. Si la lista es pequeña y conocemos n entonces nos bastaría con ir a random.org para que nos diga que elemento seleccionar. Pero si la lista es muy grande y no sabemos el tamaño de la lista entonces el problema deja de ser trivial. Es por ello que muchas veces se usan algoritmos como el que se propone a continuación:

beginitemize

- Guarde el primer elemento.
- Cuando llegue al i-esimo elemento ($i \geq 2$), con probabilidad 1/i guarde el nuevo elemento y descarte el antiguo. En caso contrario, simplemente quédese con el antiguo elemento.

  Por ejemplo, si la lista tiene 2 elementos, partirías guardando el primer elemento, y en el segundo paso cambiaríamos el elemento guardado con probabilidad 1/2.

a) Pruebe que cada elemento de la lista es elegido con probabilidad 1/n

b) Programe el algoritmo para que al recibir una lista separada por comas entregué un elemento de la lista con probabilidad n. Ejemplo:

   - Input: 1,3,5,2,1,9
   - Output:5