Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación

Tarea 1
Matemáticas Discretas

Fecha:	26 de septiembre de 2017
Autor:	Daniel Araya Poblete
Email:	Daniel.arayap1@gmail.com
Curso:	CC3101-1
Profesor:	Gonzalo Navarro

Pregunta 1.

En el país de DCCENIA se tiene un sistema monetario que utiliza monedas de valor $m_1, m_2, \dots, m_k$; con m_1= 1, $k \geq 2$ y m_i< m_{i+1} para todo $1 \leq i \leq k-1$. El Kioskito (que es una cadena de súper grande que funciona en ese país) debe dar el vuelto a todos los clientes en este sistema de monedas.

a) Demuestre que el Kioskito puede devolver cualquier vuelto X usando este algoritmo:
- Inicializar j = k.
- While True:
- if $X - m_j \geq 0$: Entregar m_j. Cambiar X por X - m_j. End if.
- if $X - m_j < 0$: Cambiar j por j - 1. End if.
- Si $X = 0$ end while.

Sea P_X la propiedad de que el vuelto X pueda ser devuelto por el Kioskito. El caso base P_1 se cumple trivialmente (una moneda m_1). Para el caso inductivo si se asume que para un $X > 1$ P_X es cierta, entonces para el caso $X + 1$ basta con entregar el vuelto para X más una moneda m_1.

b) Ahora suponga que m_i divide a $m_i + 1$ para todo $1 \leq i \leq k-1$. Demuestre que este algoritmo devuelve el vuelto en la cantidad mínima de monedas.

La cantidad de monedas que entrega el algoritmo puede expresarse como: $Q = q_k + q_{k-1} + \cdots + q_1$ con q_i la cantidad de monedas m_i que caben en el vuelto que va quedando para todo $1 \leq i \leq k$. Si decimos que $r_k = X$, entonces Q puede expresarse como: $Q = \left\lfloor \frac{r_k}{m_k} \right\rfloor + \left\lfloor \frac{r_{k-1}}{m_{k-1}} \right\rfloor + \cdots + \left\lfloor \frac{r_1}{m_1} \right\rfloor$, con $r_i = r_{i+1} \% m_{i+1}$. Como cada uno de los sub-problemas del algoritmo se resuelve entregando el mínimo (por la función piso presente en la forma de Q), entonces se tiene que el algoritmo siempre entrega la cantidad mínima de monedas.

c) De un vuelto X y una serie de valores para $m_2, m_3, \dots, m_k$ tal que el algoritmo no devuelve el vuelto en la cantidad óptima de monedas.

Para valores {1, 3, 5, 6} y un vuelto X=10, el algoritmo entregará: una moneda de 6, una moneda de 3 y una moneda de 1, cuando la solución óptima a entregar son dos monedas de 5.

d) El Kioskito tiene una demanda increíble y por tanto debe automatizar la entrega de vuelto en su sistema. Para ello suponga que m_i divide a m_i +1 y con ello implemente en Matlab un algoritmo que dado el vuelto X este diga cuantas monedas de cada tipo deben entregarse para dar el vuelto en la cantidad mínima de monedas para cualquier valores de $m_2, m_3, \dots, m_k$ (Recuerde que $m_1 = 1$).

Ver archivo P1d.java

Pregunta 2.

Para representar una relación R sobre un conjunto finito A = $a_1, \ldots, a_n$ utilizamos una matriz indicatriz, es decir, una matriz de n x n tal que A_{ij}= 1 si y solo si (a_i, a_j) está en R. Una relación R en A es cuasi orden si es refleja y transitiva.

a) Programe un código que detecte si una relación R con la representación es cuasiorden.

Ver archivo P2a.java

b) Demuestre que si R es cuasiorden entonces la mayor relación de equivalencia contenida dentro de ésta es $R \cap R^{-1}$.

Primero se demostrará que $R \cap R^{-1}$ es una relación de equivalencia:

- Sea a en A. Como sabemos que (a,a) pertenece a R, entonces (a,a) pertenece a R^{-1} y por lo tanto también a $R \cap R^{-1}$.→ $R \cap R^{-1}$ refleja.
- Sea (a,b) en $R \cap R^{-1}$. Entonces (a,b) pertenece a R y a R^{-1}, por lo tanto (b,a) pertenece a R^{-1} y a R respectivamente.→ $R \cap R^{-1}$ simétrica.
- Sean (a,b) y (b,c) en $R \cap R^{-1}$. Como R es cuasiorden, (a,c) pertenece a R. Como $R \cap R^{-1}$ es simétrica, (b,a) y (c,b) están en $R \cap R^{-1}$, entonces (c,a) pertenece a R y (a,c) pertenece a R^{-1}. Entonces (a,c) pertenece a $R \cap R^{-1}$.→ $R \cap R^{-1}$ transitiva.

∴ $R \cap R^{-1}$ es una relación de equivalencia■

Ahora, se debe hacer notar que $R \cap R^{-1} \subseteq R$. En efecto, sea $(a,b) \in R \cap R^{-1}$ $\rightarrow (a,b) \in R \wedge (a,b) \in R^{-1} \rightarrow (a,b) \in R$■

Finalmente, para demostrar que $R \cap R^{-1}$ es la máxima relación de equivalencia, se procederá por contradicción: suponga que $R \cap R^{-1}$ no es la máxima relación de equivalencia, entonces existe R' que es la máxima relación de equivalencia contenida en R. Así, $R \cap R^{-1} \subseteq R' \subseteq R$, con $R' \not\equiv R \cap R^{-1}$. Bajo esta lógica existe un par (a,b) en R' tal que (a,b) no está en $R \cap R^{-1}$. Desarrollando:

$(a,b) \in R' \Rightarrow (b,a) \in R'$ //porque R' es simétrica

$\Rightarrow (b,a) \in R$ //porque R' es subconjunto de R

$\Rightarrow (a,b) \in R^{-1}$ //por definicion de inverso de R

Y además $(a,b) \in R$ porque $(a,b) \in R'$ y $R' \subseteq R$.

$\Rightarrow (a,b) \in R \wedge (a,b) \in R^{-1}$

$\Rightarrow (a,b) \in R \cap R^{-1}$//pero $(a,b) \notin R \cap R^{-1}$

→← Contradicción

∴ $R \cap R^{-1}$ es la máxima relación de equivalencia contenida en R.■

c) Programe un código que devuelva la mayor relación de equivalencia que está contenida en una relación que es cuasi orden.

Ver archivo P2c.java

Pregunta 3

Definimos la secuencia de Fibonacci mediante la recursión:

$$\begin{cases} F_0 = F_1 = 1 \\ F_n = F_{n-1} + F_{n-2}\,; \quad n > 1 \end{cases}$$

a) Demuestre que todo entero positivo puede ser descompuesto en una suma de números de Fibonacci en la que no hay dos números de Fibonacci consecutivos.

Para la demostración de este problema se usará inducción fuerte. Se puede observar que para valores más pequeños de n esto se cumple ya que se representan como números de Fibonacci (ej. 1, 2, 3) La HI es que $\forall n \leq k,$ n puede ser descompuesto en una suma de números de Fibonacci sin que haya dos números de Fibonacci consecutivos. Ahora el paso inductivo k+1. Se tienen dos casos:

- Si k+1 es un número de Fibonacci, bingo.
- Si no, entonces k+1 está entre dos números de Fibonacci, llámense F_i y F_{i+1}. Ahora se considera un valor $p = k + 1 - F_i$. Entonces, $p < k + 1$ y por lo tanto p tiene una representación en números de Fibonacci.

  Como se expresó inicialmente, $F_i < k + 1 < F_{i+1}$ pero $k + 1 = p + F_i$ y aprovechando la forma de la secuencia Fib tenemos que $F_{i+1} = F_i + F_{i-1}$. Así se llega a $p + F_i < F_i + F_{i-1} \rightarrow p < F_{i-1}$.

  $\therefore$ F_{i-1} no pertenece a la representación en números de Fib de p y por lo tanto se puede tomar $k + 1 = p + F_i$ como la representación en números de Fib de k+1 que no tiene dos números de Fibonacci consecutivos.■

b) Consideramos que la descomposición encontrada en el punto anterior es única. Programe una función fibcode(n) en Matlab que reciba un entero positivo n y retorne un número binario que cumpla con la siguiente regla: el k-ésimo dígito de derecha a izquierda es 1 ssi F_k está en la descomposición de Fibonacci de n. Esta descomposición debe cumplir con que no se permite dos números de Fibonacci consecutivos.

Para esta pregunta se consideró que el dígito inicial de la representación es 1, ya que no nos interesa usar F_0 en la representación del número (si en la descomposicion del entero se utiliza F_1 entonces la representación de éste terminará en '11', que no cumple la propiedad de (a)).

Ver archivo P3b.java