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

Tarea 2: Combinatoria y Probabilidades

Fecha:	05 de junio de 2017
Autor:	Daniel Araya Poblete
Email:	Daniel.arayap1@gmail.com
Curso:	CC3101-1
Profesor:	Pablo Barceló

Pregunta 1.

Consideremos dos funciones recursivas:

$$f(0) = 1$$
$$g(0) = 2$$

$$f(n) = \big(5f(n-1) + 2g(n-1)\big)mod(100)$$
$$g(n) = \big(3f(n-1) + g(n-1)\big)mod(100)$$

Queremos calcular $\sum_{i=0}^{k}(f(i) + g(i))$.

a) Use un argumento de palomar para demostrar que $\exists n_0, n_1 \;\; f(n_0) = f(n_1) \wedge g(n_0) = g(n_1)$. Use esto para concluir que las funciones son periódicas a partir de algún n.

Supongamos por contradicción que esto no se cumple, es decir, $\forall n_0, n_1 \, f(n_0) \neq f(n_1) \vee g(n_0) \neq g(n_1)$. Esto significa que f(n) y g(n) toman infinitos valores distintos en ℕ. Pero las funciones f(n) y g(n) tienen la forma $f(n) = \big(af(n-1) + bg(n-1)\big)mod(100)$, por lo tanto, *f* y *g* pueden tomar como máximo 100 valores distintos. Entonces, por principio del palomar, si *f* y *g* entregan 101 resultados, al menos uno de ellos se repetirá una vez.

Sabemos que $f(n > 99)$ está en el arreglo (independiente de los parámetros $f(n-1)$ y $g(n-1)$, porque al aplicar mod (100) el resultado será necesariamente un valor entre 0 y 99), por lo tanto, podemos buscarlo, recorriendo el arreglo desde $a[0], .., a[99]$ y si $i = n\%100, f(n) = a[i]$.

Esto último puede demostrarse por inducción, tomando como caso base $f(100) = f(0) = 1$, entonces f(101)=5f(100)+2g(100)=5f(0)+2g(0)=f(1). Así, asumiendo que **existe algún n** tal que $f(100+n) = f(n)$, se realiza el paso inductivo $f(100+n+1) = 5f(100+n) + 2g(100+n) = 5f(n) + 2g(n) = f(n+1)$ y se demuestra la periodicidad.

b) Usando la parte a) encuentre un algoritmo para calcular $\sum_{i=0}^{k}(f(i) + g(i))$ para un k cualquiera. Comente sobre la complejidad de este algoritmo.

- **Calcular f(i)**: crear un arreglo de 100 valores para guardar los primeros 99 datos (desde 0). Se guarda el caso base f(0)=1. Luego, si el valor de la casilla k es 0 en el arreglo, ésta se reemplaza por $(5f(i-1) + 2g(i-1))mod(100)$ hasta que se rellene el arreglo. Para valores mayores a 99 simplemente se aplica mod(100) y se busca en el arreglo su valor ya que éste existe en el arreglo como se demostró en la parte a).
- **Calcular g(i)**: de forma similar, sólo cambia la forma de la función.
- **Calcular h(i)**: parra cualquier i, se retorna f(i%100)+g(i%100)
- **Calcular la sumatoria(k)**: igual que antes, se crea un arreglo c de tamaño 100. Luego se añade el caso base c(0)=3. Mediante desarrollo iterativo se guardan los valores de los primeros 100 elementos haciendo que c(i)=c(i-1)+h(i). Para k=123, la función debe retornar 1*c[99]+c[23]. Generalizando esto, la función debe retornar *c*[k%100]+k/100**c*[99], siendo k/100 la cantidad de veces que se calcula la sumatoria de los primeros 100 elementos, y a esto se suma el valor c[k%100].

c) Escriba un programa que reciba un k y calcule $\sum_{i=0}^{k}(f(i)+g(i))$. Como el valor puede ser muy grande entréguelo módulo $10^9 + 7$.

Ver archivo P1c.py.

d) Demuestre que independiente de cual sea el módulo que se le aplique a las funciones, éstas siempre van a ser periódicas a partir de algún punto. Es decir que el 100 se puede reemplazar por otro valor que no tiene que ser el mismo para ambas como aparece más abajo.

En esta parte se cumple el mismo supuesto que en la parte a, es decir, para ambas funciones el dominio de la función es infinito en los números naturales, pero el recorrido es acotado. Sean m y l los módulos aplicados a f y g respectivamente, tenemos que f entonces puede entregar m-1 valores distintos y g entrega l-1 valores. Entonces usando el principio del palomar se tiene que a partir de cierto n los valores comenzarán a repetirse, i.e. las funciones son periódicas. El resultado de f(i)+g(i) por lo tanto es también una función periódica, ya que el dominio de ésta sigue siendo $\mathbb{N}$ y el recorrido va de 3 a (m+l-2), es decir, acotado. Entonces, si los módulos cambian (o si las constantes que multiplican a f y g cambian), la función seguirá siendo periódica.

e) Generalice su solución, de tal forma que reciba además de k los parámetros a, b, c, d, m, l y que resuelva el problema para:

$$f(n) = \big(af(n-1) + bg(n-1)\big)mod(m)$$
$$g(n) = \big(cf(n-1) + dg(n-1)\big)mod(l)$$

Pregunta 2.

a) Encuentre una fórmula para calcular la probabilidad de que en una partida con un jugador aleatorio X salga ganador. Todos los jugadores tienen la misma probabilidad de salir elegidos. Como ejemplo imagine que $h_X = 10$ y hay otros 2 jugadores ($J = 2$) separados en 2 grupos y que $G_1 = 1$; $h_1 = 0$, $G_2 = 1$; $h_2 = 40$. Esto quiere decir que hay un jugador de habilidad 0 y uno de habilidad 40. En este caso la probabilidad de que A gane es 0,6.

Para entender el ejemplo, se tiene que la probabilidad de que X gane es $\Pr(X) = \frac{h_X}{h_X+h_i}$ siendo i={1,2}. Pero como se tienen 2 posibles contrincantes, $\Pr(X) = \Pr(X|jugar\ con\ el\ primero) + \Pr(X|jugar\ con\ el\ segundo)$, entonces

$$\Pr(X) = \frac{h_X}{h_1 + h_X} * \frac{1}{2} + \frac{h_X}{h_2 + h_X} * \frac{1}{2} = 0{,}6$$

Pero en este caso, J=2 y se tenían sólo dos grupos G_0 y G_1 con un jugador cada uno. Para el caso en que $G_i > 1$, la probabilidad de jugar con un miembro de habilidad h_i será $\frac{G_i}{J}$. Esto se puede generalizar como sigue:

$$\Pr(X) = \sum_{i=1}^{k} \frac{h_X}{h_X + h_i} * \frac{G_i}{J} = \frac{h_X}{J} * \sum_{i=1}^{k} \frac{G_i}{h_X + h_i}$$

b) Escriba una función que reciba h_X, J y la información sobre los jugadores que hay disponibles (los G_i) y calcule la probabilidad de que al enfrentarse con un jugador al azar gane X. El input se entregará en dos líneas, en la primera se entregarán los enteros h_X , J y k. Luego se entregarán k líneas, en la línea i se encontrarán G_i y h_i separados por espacios.
Ejemplo:
10 2 2
1 0
1 40
Resultado:
0.6

Ver archivo P2b.py

c) Ahora calcularemos la probabilidad de que X gane el premio. Considere que X va a jugar m partidas y que cada vez que juega con alguien ese jugador sigue disponible, incluso pueden volver a enfrentarse (en otras palabras, es un juego con reposición). Describa un algoritmo que permita determinar la probabilidad de que X gane el premio, asumiendo que cada vez que juega, el adversario es elegido al azar de manera equiprobable entre los demás jugadores. Luego muestre su correctitud. Puede utilizar la fórmula de la parte a). Este algoritmo debe ejecutarse en tiempo polinomial sobre m, argumente porqué su algoritmo efectivamente cumple con esa condición.

d) Escriba un programa que calcule la probabilidad de que X gane el premio. Éste debe recibir toda la información relevante en el mismo formato que la pregunta anterior, agregando una primera línea con n y m. El programa debe entregar el resultado como un número con a lo menos 3 dígitos de precisión después de la coma.
Se entrega en la primera línea los enteros n y m y en la segunda h_X, J y k separados por espacios. Luego en las siguientes k líneas se entregarán los enteros G_i y h_i.
Ejemplo:
2 4
10 5 3
2 0
2 10
1 20
Resultado:
0,641