Nombre y Apellido: ........................................................................................... N° Legajo: .................

#### Segundo Parcial de Estructuras de Datos y Algoritmos

#### Segundo Cuatrimestre de 2016

|  |  |  |  |
| --- | --- | --- | --- |
| Ejercicio 1 | ***Ejercicio 2*** | ***Ejercicio 3*** | Nota |
|  |  |  |  |

Condición Mínima de Aprobación: Tener por lo menos dos ejercicios con B-

## Ejercicio 1

## Basándonos en la implementación de grafos con lista de adyacencia vista en clase, se desea, dado un grafo acíclico dirigido, obtener el camino de mayor peso entre 2 nodos. La implementación de la solución debe ser iterativa.

**public int maxPathIterative(V from, V to);**

## Ejercicio 2

En el mundo de las finanzas y las monedas es posible especular con los cambios de valor de intercambio que estas pueden ir teniendo a lo largo del tiempo.

Se cuentan con N distintas monedas y M meses para cambiar dichas monedas. En cada mes podemos cambiar de moneda (o no) solo una vez.

Se pide:

1. Modelar el problema como un grafo dirigido en el cual cada nodo corresponde a un mes en el tiempo con una moneda determinada, y cada arista entre nodo y nodo significa el pasaje de moneda en el tiempo. Naturalmente solo habrán aristas en un solo sentido y estas tendrán un multiplicador que corresponde al cambio de moneda, como se ilustra en el ejemplo
2. Implementar un método que reciba un número de plata inicial X y que recorra el grafo haciendo los cambios de moneda óptimos a fin de maximizar el valor final de dinero.

Ejemplo de grafo de intercambio de moneda



**public class ExchangeCurrencyGraph {**

**// modelado**

**public double computeMaxCurrencyExchange() {**

**// Implementación**

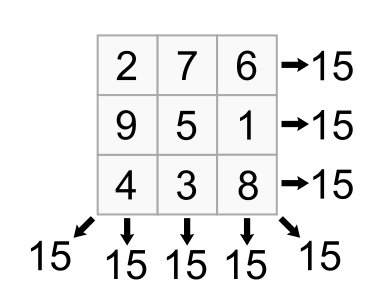
**}**

**}**

## Ejercicio 3

Un cuadrado mágico de orden N es una matriz de NxN llena con números del 1 al N en el cual se cumple la propiedad que la suma de los elementos de cada fila, cada columna y las dos diagonales son iguales.

Ejemplo:



Se pide implementar un algoritmo que dado un N devuelva un cuadrado mágico de orden N

**public int[][] magicSquare(int n);**