Programación Avanzada
Trabajo Practico Nº 3
2º cuatrimestre 21016 (ambos turnos)

DIIT - Universidad Nacional de La Matanza

05/09/2016

1. Introducción

A través del presente trabajo se espera que los alumnos codifiquen, evalúen y comparen distintos algoritmos que resuelven, mediante diferentes técnicas, el mismo problema.
El problema a resolver es la codificación de un algoritmo que evalué un polinomio de grado n.

2. Objetivo

Diseñar un programa para evaluar un polinomio P(x) de grado n.

```
public class Polinomio {
  private int grado;
  private double[] coeficientes;
   //La posicion 0 del arreglo de coeficientes contiene el coeficiente de grado n y la
        posicion n contiene al termino independiente.

  public Polinomio{...}

    double evaluarMSucesivas (double x){...}
    double evaluarRecursiva (double x){...}
    double evaluarRecursivaPar (double x){...}
    double evaluarProgDinamica (double x){...}
    double evaluarMejorada (double x){...}
    double evaluarPow(double x){...}
    //y a sugerencia de Lucas P
    double evaluarHorner( double x){...}

  }
```

1. Escribir evaluarMSucesivas utilizando cálculo de potencia por multiplicaciones sucesivas
2. Escribir evaluarRecursiva utilizando el siguiente cálculo de potencia recursiva:
   a) Sin considerar si el exponente es par o impar:

   ```
   potencia(x,n)=x*potencia(x,n-1)
   ```

   b) Considerando si el exponente es par o impar:
   - Si n es par:

     ```
     potencia(x,n)=potencia(x*x,n/2)
     ```

   - Si n es impar

     ```
     potencia(x,n)=x*potencia(x,n-1)
     ```

3. Escribir evaluarProgDinamica, almacenando las potencias de X ya calculadas.
4. Escribir evaluarMejorada, con un algoritmo de igual complejidad computacional que el anterior, pero que ejecute en un tiempo menor.

5. Escribir evaluarPow, valiendose del metodo Math.pow(x,n) provisto por el lenguaje Java. Se debe incluir dentro de alguno de los metodos anteriores donde se considere que es apropiado. Investigue la CC de Math.pow.

6. Escribir evaluarHorner, aplicando el algoritmo de Horner de análisis numérico. (Investigar)

3. Análisis de complejidad computacional

Indique la funcion de complejidada computacional O asociada a cada uno de los metodos implementados.

4. Gráficos y tablas de rendimiento comparativo

Compare el tiempo de ejecución de los cuatro métodos evaluar() para un mismo polinomio.
Repita este procedimiento para distintos valores del grado.
Genere todos los polinomios que considere necesarios para realizar el análisis.

5. Conclusiones

A partir del análisis comparativo extraiga conclusiones.