INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO

Análisis de Algoritmos

Sesión 6: Matriz de Vandermonde

Edgar Adrián Nava Romo

Maestra: Sandra Díaz Santiago
Grupo: 3CM3

Manual de Usuario

Dependencias:

Python =< 3.7
Pandas
xlrd
openpyxl
cmath
numpy

Para correr el programa usar comando:

$ python Vandermonde.py

Input: Modificar el Archivo *Factores.txt*, cada línea es un Factor

Output: Se generan 6 Documentos “xlsx” para usar los datos por separado o bien, se puede ver el archivo *Final.xlsx* para ver todos los resultados, para leer este archivo se cuenta con el siguiente orden:

Matrix2.xlsx	Los factores ingresados por el Usuario
Vandermore.xlsx	La matriz Vandermonde Generada por el programa
Inverse.xlsx	La inversa de la Matriz Vandermonde
Identity.xlsx	Comprobación de la Inversa con la Matriz Vandermonde
Coeficients.xlsx	Coeficientes Finales
Final.xlsx	Archivo que cuenta con todos los archivos anteriores concatenados

Código Fuente

Librerías Usadas:

```
1.  import sys
2.  import cmath
3.  import numpy
4.  import pandas as pd
5.  from pandas import *
```

Función que genera la matriz de Vandermonde con las raíz principal dada en main.

```
6.  def Vandermonde(vector_x):
7.      vector_x
8.      vandermonde_matrix = []
9.      pow = len(vector_x) -1
10.     for element in vector_x:
11.         vandermonde_matrix.append([])
12.         for index in range(pow+1):
13.             vandermonde_matrix[-1].append(element[0]**index)
14.     return vandermonde_matrix
```

Función que determina la Inversa de la matriz de Vandermonde Generada por medio de su Determinante (solo se muestra código principal de la Inversa)

```
15. def MatrixInverse(m):
16.     determinant = Determinant(m)
17.     #special case for 2x2 matrix:
18.     if len(m) == 2:
19.         return [[complex(numpy.around(m[1][1]/
    determinant,4)), complex(numpy.around(-1*m[0][1]/determinant,4))],
20.         [complex(numpy.around(-1*m[1][0]/
    determinant,4)), complex(numpy.around(m[0][0]/determinant,4))]]
21.     #find matrix of cofactors
22.     cofactors = []
23.     for r in range(len(m)):
24.         cofactorRow = []
25.         for c in range(len(m)):
26.             minor = MinorMatrix(m,r,c)
27.             cofactorRow.append(   complex(numpy.around(((-1)**(r+c)) *
    Determinant(minor), 4))   )
28.         cofactors.append(cofactorRow)
29.     cofactors = Transpose(cofactors)
30.     for r in range(len(cofactors)):
31.         for c in range(len(cofactors)):
32.             cofactors[r][c] = cofactors[r][c]/determinant
33.     return cofactors
```

Función para Multiplicar Matrices, con ella se sacan la matriz Identidad por la Regla *AB = BA = I* así como los coeficientes, multiplicando los factores x la matriz de Vandermonde

```
34. def Mult_Matrix(matrix1, matrix2):
35.     matrix1_n_rows = len(matrix1)
36.     matrix1_n_columns = len(matrix1[0])
37.     matrix2_n_rows = len(matrix2)
38.     matrix2_n_columns = len(matrix2[0])
39.     if matrix1_n_columns != matrix2_n_rows:
40.         print("{} vs {}".format(matrix1_n_columns, matrix2_n_columns))

41.         print("Not compatible Matrix\n")
42.         sys.exit()
43.     result_matrix = []
44.     # Creates a null matrix with the resultant dimensions
45.     for column in range(matrix1_n_rows):
46.         result_matrix.append([])
47.         for row in range(matrix2_n_columns):
48.             result_matrix[column].append(0)
49.     # Mult_Matrix
50.     for i in range(matrix1_n_rows):
51.         for k in range(matrix2_n_columns):
52.             for j in range(matrix2_n_rows):
53.                 result_matrix[i][k] += complex(numpy.around(matrix1[i]
    [j] * matrix2[j][k] , 4))
54.     return result_matrix
```

En Main primero se lee un Archivo con los factores a valorar, cabe destacar que si el número de factores no es potencia de 2^k la matriz de Vandermonde será la última potencia de 2^k que se hizo pero no se hará la multiplicación, se cuenta el número de estos y se hace el procedimiento. Con la Librería Pandas las listas fueron convertidas en 'xlsx' para facilitar la lectura al usuario.

```
55. if __name__ == '__main__':
56.     x = [] #Vandermonde
57.     with open('A.txt', 'r') as f:
58.         y = [[complex(digit) for digit in line.split()] for line in f]

59.     #print(y)
60.     n = int(1 << (len(y) - 2).bit_length())
61.     base = n ** (1.0/2)
62.     for i in range(0 , n):
63.         x.append([complex(cmath.cos(2*cmath.pi * i / n), cmath.sin((2*c
    math.pi * i) / n))])
64.
65.     p = Vandermonde(x)
66.     print("Vandermore is Ready\n")
67.
68.     pIn = MatrixInverse(p)
69.     print("Inverse is Ready\n")
70.
71.     pId = Mult_Matrix(pIn,p)
72.     print("Identity is Ready\n")
73.
```

```
74.     coeficients = Mult_Matrix(p, y)
75.
76.     print('Coefficients are:\t')
77.     for i in range(0 , n):
78.         print("x^{}".format(n-i-1) , coeficients[i], end = ' + ')
79.     pd.DataFrame(y).to_excel('Matrix2.xlsx', header='Matrix2')
80.     pd.DataFrame(p).to_excel('Vandermore.xlsx', header='Vandermore')
81.     pd.DataFrame(pIn).to_excel('Inverse.xlsx', header='Inverse')
82.     pd.DataFrame(pId).to_excel('Identity.xlsx', header='Identity')
83.     pd.DataFrame(coeficients).to_excel('Coeficients.xlsx', header='Coef
    icients')
84.
85.     excel_names = ["Matrix2.xlsx", "Vandermore.xlsx", "Inverse.xlsx", "
    Identity.xlsx",
86.     "Coeficients.xlsx"]
87.     # read them in
88.     excels = [pd.ExcelFile(name) for name in excel_names]
89.     # turn them into dataframes
90.     frames = [x.parse(x.sheet_names[0], header=None,index_col=None) for
     x in excels]
91.     # delete the first row for all frames except the first
92.     # i.e. remove the header row -- assumes it's the first
93.     frames[1:] = [df[1:] for df in frames[1:]]
94.     # concatenate them..
95.     combined = pd.concat(frames)
96.     # write it out
97.     combined.to_excel("Final.xlsx", index = None)
```

Capturas de Pantalla

Las pruebas finales se pueden encontrar en la carpeta “Pruebas” dentro del archivo .zip entregado junto con la práctica

Bibliografía Consultada

https://stackoverflow.com

https://en.wikipedia.org/wiki/Vandermonde_matrix

https://mathworld.wolfram.com/VandermondeMatrix.html

https://en.wikiversity.org/wiki/Numerical_Analysis/Vandermonde_example

http://www-users.math.umn.edu/~garrett/m/algebra/notes/17.pdf