Shepley Algorithm implemented in Python

Author Eng. Sergio Adrián Martin

Introduction

Shepley's method is one of several methods devised to invert matrices.
The advantage it has is that it is not recursive and does not require more than an array.
I have known it for about 40 years and although it is not my favorite method and I don't know who Shepley was, I recognize that it can be implemented in a variety of languages, and this time in Phyton.

The essence of the Algorithm

The algorithm is based on a set of rules.
1. An element of the main diagonal a(I,I) is chosen.
If a(I,I) is different from 0, divide all elements in row I by -A(I,I) except A(I,I).
3. Each element a(j,k) where j is different from I, and k different from I is affected by the product of a(I,k) by a(j,I).
A(j,k)<-a(j,k)+a(i,k)a(j,i)
4. Each element in column I is divided by -a(I,I).
5. The element a(I,I) is replaced by -1/a(I,I).
6. The process is repeated for the next value of I as long as a(I,I) is not 0. In case it is, you can jump to another value of I as long as at some point you iterate using all the elements of the main diagonal as a base.
7. The result will contain in A all the elements of the inverse but with negative sign.
Of course if a matrix has a determinant value of 0 it will not have an inverse.

Below is the source code

```
# using shepley to invert matrix
#done by Sergio Martin
import array as arr
import numpy as np
import math
na="1"
while na!="":
    na=input("number of rows")
    if na=="":
        break
    n=int(na)
    #a = arr.array('i',[0,0])
    a=np.ones((n,n))
    print("enter the elements")
    for i in range(0,n,1):

        for j in range(0,n,1):
            s="a"+str(i+1)+","+str(j+1)+"="
            a[i][j]=int(input(s))
    #investment
    for i in range(0,n):
```