Autor: Paweł Salwa

Program ten ma złożoność liniową O(n)- jest tylko kilka pojedynczych pętli w funkcjach.

wynik:
x0= -0.2601626016260163
x1= 0.44715447154471555
x2= 0.4715447154471545
x3= 0.6666666666666669
x4= 0.8617886178861788
x5= 0.8861788617886177
x6= 1.5934959349593498

Kod w pythonie:

```
#================================================================
def shermanMorrison(Z, Q ,UV):
	global N

	mian = 0.
	liczn = 0.

	for i in range(0,N):
		liczn += Z[i]*UV[i]
		mian += Q[i]*UV[i]

	mian += 1

	for i in range(0,N):
		Q[i] *= (liczn/mian)


#================================================================
def thomas(A,B,C,D):
	global N

	N -= 1

	C[0] /= B[0]
	D[0] /= B[0]

	for i in range(1,N):
		C[i] /= B[i] - A[i]*C[i-1]
		D[i] = (D[i] - A[i]*D[i-1]) / (B[i] - A[i]*C[i-1])

	D[N] = (D[N] - A[N]*D[N-1]) / (B[N] - A[N]*C[N-1])


	i = N - 1
	while i >= 0:
		D[i] -= C[i] * D[i+1]
```

```
        i -= 1

#===============================================================

A = [0.,1.,1.,1.,1.,1.,1.] #wedlug wzoru (60) z wykladu
B = [3.,4.,4.,4.,4.,4.,3.] #thomasem uzyskuje wektory
C = [1.,1.,1.,1.,1.,1.,0.] #uzyte we wzorze shermanamorrisona

Z = [1.,2.,3.,4.,5.,6.,7.]
UV = [1.,0.,0.,0.,0.,0.,1.]
Q = [1.,0.,0.,0.,0.,0.,1.]
N = len(A)
thomas(A,B,C,Z)

A = [0.,1.,1.,1.,1.,1.,1.]#definiuje jeszcze raz bo thomas() mi zmienia
B = [3.,4.,4.,4.,4.,4.,3.]
C = [1.,1.,1.,1.,1.,1.,0.]
N = len(A)
thomas(A,B,C,Q)

A = [0.,1.,1.,1.,1.,1.,1.]#definiuje jeszcze raz bo thomas() mi zmienia
B = [3.,4.,4.,4.,4.,4.,3.]
C = [1.,1.,1.,1.,1.,1.,0.]
N = len(A)
shermanMorrison(Z, Q, UV)

N = len(A)
print 'wynik:'
for i in range(0,N):
    print "x" + repr(i) + "=  "+repr(Z[i]-Q[i])
```