In [17]:
```
import numpy as np
import scipy
import matplotlib.pyplot as plt
import math
print("Hello World")
```

```
Hello World
```

In [18]:
```
#Generating a random matrix
M = np.random.rand(100,100)

#Making it symmetric
A = M  + M.T
```

In [19]:
```
#Finding the eigenvalues and eigenvectors for the matrix and sorting them in a
eig = np.linalg.eigvals(A)
idx = np.argsort(eig)
eig = eig[idx]
```

In [20]:
```
print(eig[0:10])
#Thus, the smallest eigenvalue is roughly - 8
```

```
[-7.49050678 -7.37120721 -7.055183   -6.85956998 -6.68348424 -6.51323141
 -6.44438292 -6.12730853 -6.0315539  -5.78679985]
```

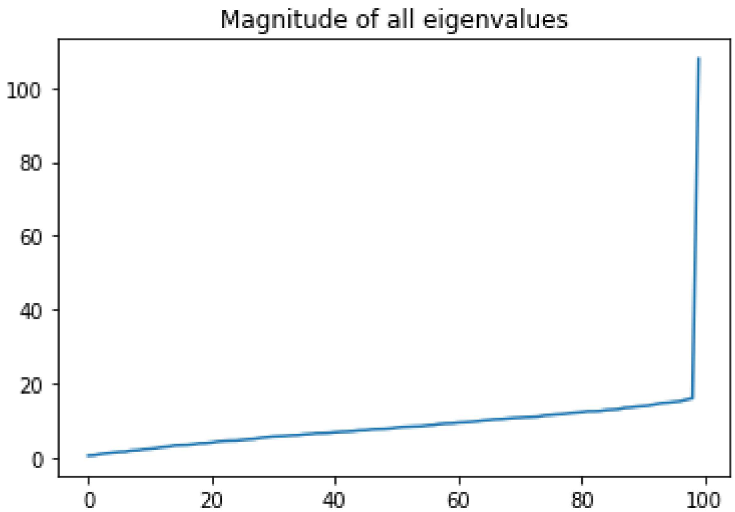
In [21]:
```
B = A + 8*np.eye(100,100)#Making a new, symmetric, positive eigenvalue matrix
```

In [22]:
```
eigB = np.linalg.eigvals(B)
idxB = np.argsort(eigB)
eigB = eigB[idxB]
```

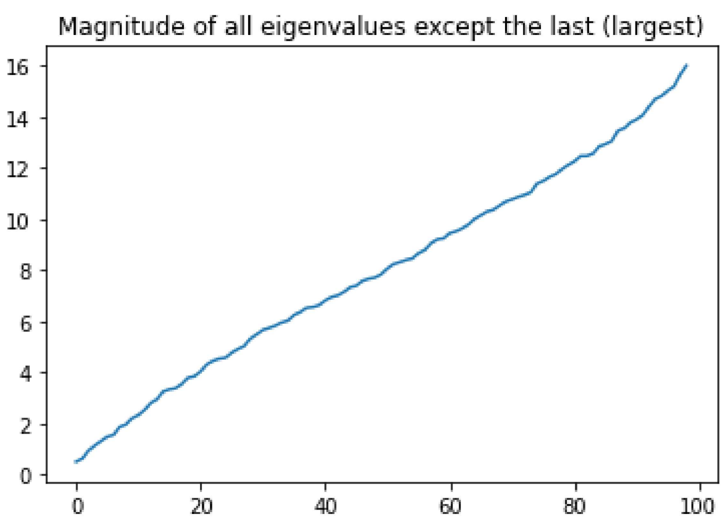
In [23]:
```
#Thus, B has only positive eigenvalues and is symmetric
print(eigB[0:10])
print(eigB[90:])
```

```
[0.50949322 0.62879279 0.944817   1.14043002 1.31651576 1.48676859
 1.55561708 1.87269147 1.9684461  2.21320015]
[ 13.91333665  14.0829986   14.41607486  14.71230028  14.83131923
  15.03304404  15.20856989  15.66580165  16.01998977 107.9233933 ]
```

In [24]:
```
#There's a sudden spike In Eigenvalues, plotting it here
plt.plot(eigB)
plt.title("Magnitude of all eigenvalues")
plt.show()
```



In [25]:
```
#Plotting without the last eigenvalue
plt.plot(eigB[0:99])
plt.title("Magnitude of all eigenvalues except the last (largest)")
plt.show()
```



In [26]:
```
#Picking sigma to be uniformly distributed between 3 and 11
sigma = np.linspace(3,10,8)
print(sigma)
```

```
[ 3.  4.  5.  6.  7.  8.  9. 10.]
```

In [27]:
```
#Generate the random vector
s = np.random.normal(0, 1, 100)

norm=math.sqrt(sum(s*s))
S=s/norm
```

In [28]:
```
#Creating Vi
V = np.zeros((8,100))
for i in range(8):
    V[i] = np.dot(np.linalg.inv(sigma[i]*np.eye(100) - B),S)
```

In [29]:
```
#Creating the H and S matrices
Hmatrix = np.ones((8,8))
Smatrix = np.ones((8,8))

for i in range(8):
    for j in range(8):
        Hmatrix[i,j]= np.dot(V[i],np.dot(B,V[j]))
        Smatrix[i,j] = np.dot(V[i],V[j])

print(Hmatrix.shape)
print(Smatrix.shape)
```

```
(8, 8)
(8, 8)
```

In [30]:
```
eigvals, eigvecs = scipy.linalg.eigh(Hmatrix, b = Smatrix,eigvals_only=False)
```

In [31]:
```
print("Eigenvalues from Filter Diagonalization: ", eigvals)
print("Sigma values: ", sigma)
```

```
Eigenvalues from Filter Diagonalization:  [2.97397875 4.02097334 5.01184986
6.0251865  6.99846498 8.08589617
 9.04534406 9.98174954]
Sigma values:  [ 3.   4.   5.   6.   7.   8.   9. 10.]
```

In [32]:
```
print(eigB)
#List of all eigenvalues of the matrix
```

```
[  0.50949322   0.62879279   0.944817     1.14043002   1.31651576
   1.48676859   1.55561708   1.87269147   1.9684461    2.21320015
   2.33409813   2.53729997   2.80837679   2.95680091   3.25922472
   3.34021554   3.38816053   3.55410613   3.79996364   3.85173042
   4.04265721   4.31738901   4.46000193   4.54294836   4.57750192
   4.77780494   4.92057056   5.04191205   5.33257578   5.49543185
   5.6716975    5.747406     5.83842234   5.96140596   6.0308938
   6.25402174   6.37036122   6.536283     6.56589273   6.6365373
   6.82406591   6.95615608   7.02125754   7.15593838   7.3391916
   7.40284814   7.5954636    7.67454812   7.72132611   7.85169538
   8.07765773   8.2473632    8.32325089   8.40800869   8.46606169
   8.67389203   8.79739554   9.06372457   9.21512087   9.25287449
   9.45537939   9.52792909   9.64384484   9.79478391  10.01711295
  10.15687417  10.30662349  10.37262679  10.54202563  10.69756441
  10.78239719  10.87955119  10.95198356  11.05817202  11.39881848
  11.50375334  11.65804613  11.76828614  11.96212877  12.13119319
  12.26519907  12.47978768  12.48468629  12.57611058  12.85830105
  12.94486207  13.06702238  13.46783201  13.5627263   13.78665256
  13.91333665  14.0829986   14.41607486  14.71230028  14.83131923
  15.03304404  15.20856989  15.66580165  16.01998977 107.9233933 ]
```

Conclusions

Nature of Convergence

- One of the sigma's was 7, and the closest eigenvalue was 7.02, but the algorithm seemed to converge towards the next nearest - 6.956. Why?

Type *Markdown* and LaTeX: α^2