In [28]:
```
import numpy as np
import scipy
import matplotlib.pyplot as plt
import math
print("Hello World")
```

```
Hello World
```

In [15]:
```
#Generating a random matrix
M = np.random.rand(100,100)

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

In [16]:
```
#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 [17]:
```
print(eig[0:10])
#Thus, the smallest eigenvalue is roughly - 8
```

```
[-7.76511461 -7.41250872 -7.06098884 -7.0291821  -6.52435987 -6.39746043
 -6.18038715 -5.83043277 -5.64613713 -5.54494272]
```

In [18]:
```
B = A + 8*np.eye(100,100)#Making a new, symmetric, positive eigenvalue matrix
```

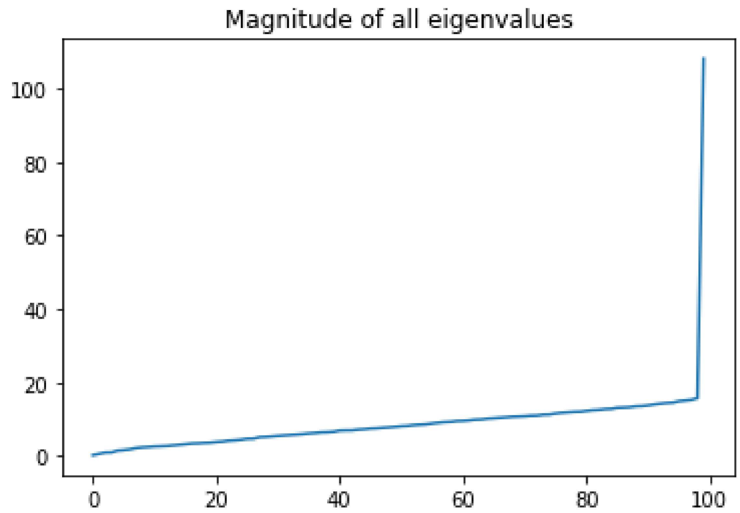
In [19]:
```
eigB = np.linalg.eigvals(B)
idxB = np.argsort(eigB)
eigB = eigB[idxB]
```

```
[0.23488539 0.58749128 0.93901116 0.9708179  1.47564013 1.60253957
 1.81961285 2.16956723 2.35386287 2.45505728]
```

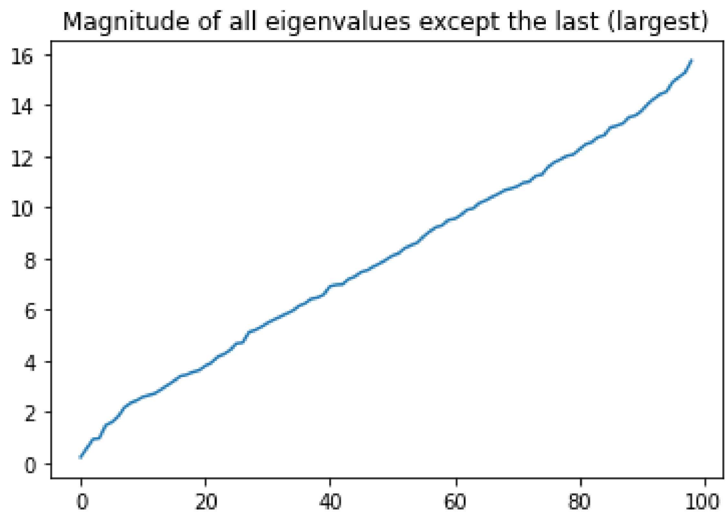
In [20]:
```
#Thus, B has only positive eigenvalues and is symmetric
print(eigB[0:10])
print(eigB[90:])
```

```
[0.23488539 0.58749128 0.93901116 0.9708179  1.47564013 1.60253957
 1.81961285 2.16956723 2.35386287 2.45505728]
[ 13.77954459  14.04774701  14.25817847  14.432823    14.53052737
  14.91433547  15.0981331   15.28699571  15.7456337  108.092191  ]
```

In [63]:
```
#There's a sudden spike In Eigenvalues, plotting it here
plt.plot(eigB)
plt.title("Magnitude of all eigenvalues")
plt.show()
```



In [64]:
```
#Plotting without the last eigenvalue
plt.plot(eigB[0:99])
plt.title("Magnitude of all eigenvalues except the last (largest)")
plt.show()
```



In [27]:
```
#Picking sigma to be uniformly distributed between 3 and 11
sigma = np.linspace(3,10,5)
print(sigma)
```

```
[ 3.    4.75  6.5   8.25 10.  ]
```

In [35]:
```
#Generate the random vector
s = np.random.normal(0, 1, 100)

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

In [42]:
```
#Creating Vi
V = np.zeros((5,100))
for i in range(5):
    V[i] = np.dot(np.linalg.inv(sigma[i]*np.eye(100) - B),S)
```

In [44]:
```
#Creating the H and S matrices
Hmatrix = np.ones((5,5))
Smatrix = np.ones((5,5))

for i in range(5):
    for j in range(5):
        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)
```

```
(5, 5)
(5, 5)
```

In [51]:
```
eigvals, eigvecs = scipy.linalg.eigh(Hmatrix, b = Smatrix,eigvals_only=False)
```

In [62]:
```
print("Eigenvalues from Filter Diagonalization: ", eigvals)
print("Sigma values: ", sigma)
```

```
Eigenvalues from Filter Diagonalization:  [3.0468459  4.72318795 6.54058035
8.22073418 9.97260057]
Sigma values:  [ 3.    4.75  6.5   8.25 10.  ]
```

In [59]:
```
print(eigB)
#List of all eigenvalues of the matrix
```

```
[2.59464048 2.66062351 2.7376065  2.89526705 3.05463685 3.21872261
 3.40207525 3.46017014 3.56226061 3.62856999 3.80446441 3.92990208
 4.16086869 4.26296387 4.42426268 4.6819402  4.714551   5.12327073
 5.2071709  5.33262243 5.48756614 5.61120879 5.72780214 5.84501154
 5.96010884 6.14549472 6.25178177 6.42924746 6.47637219 6.59274119
 6.91962321 6.97799534 6.99372135 7.20243628 7.30031387 7.47660426
 7.54863325 7.69326271 7.81242939 7.95094424]
[ 8.11149539  8.20240255  8.4070093   8.52033703  8.62038714  8.85917671
  9.05918823  9.22546782  9.29972617  9.50947706  9.55615692  9.7043694
  9.90791359  9.96887626 10.17699022 10.27954074 10.41375531 10.53193136
 10.67530415 10.74042348 10.82098287 10.96521001 11.01424369 11.23779304
 11.2902243  11.58541569 11.76676933 11.88597517 12.0185635  12.06886458
 12.26983178 12.47310877 12.55615322 12.75167368 12.82890683 13.12728743
 13.19751198 13.2896712  13.53088351 13.59698756 13.77954459 14.04774701
 14.25817847 14.432823   14.53052737 14.91433547 15.0981331  15.28699571
 15.7456337 ]
```

Conclusions

Nature of Convergence

- The algorithm appears to generate eigenvalues of the symmetric positive eigenvalue matrix. How do I minimize the computational error?
- One of the sigma's was 6.5, and the closest eigenvalue was 6.476, but the algorithm seemed to converge towards not the next (6.42), but the 3rd closest eigenvalue - 6.59, and terminated at 6.54. Why?
- A probable reason for the above is that nearby eigenvalues seems to confuse the algo a bit; in other cases, wherein closest eigenvalues to each sigma value were quite far apart, the FD algorithm correctly picked the nearest eigenvalue to converge at
- VERY COOL - I've begun to appreciate what the work will lead to.

Odd eigenvalue

Surprisingly the last eigenvalue was much larger than expected, as the values of the previous 99 seemed to increase smoothly. What factors can cause such sudden spikes in eigenvalue magnitude?

Type *Markdown* and LaTeX: α^2