Project 1 Report

Hoan Trna

26 April 2019

In [96]:
```
import matplotlib.pyplot as plt
import numpy as np
import math
import scipy.interpolate as ip
from sklearn.linear_model import LinearRegression
```

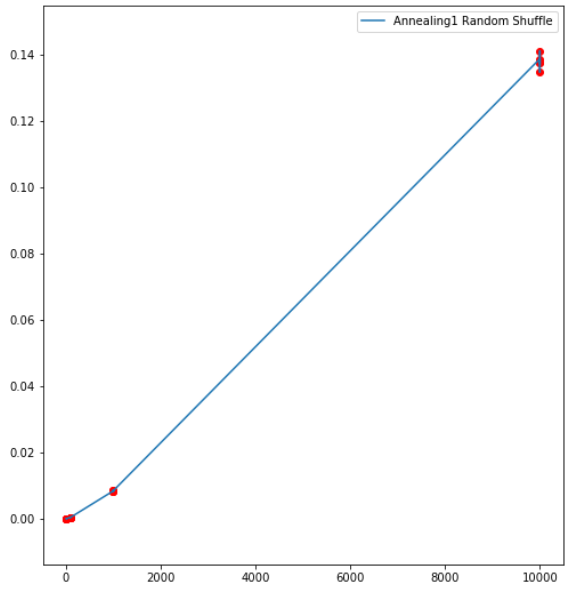
1. Annealing Sort

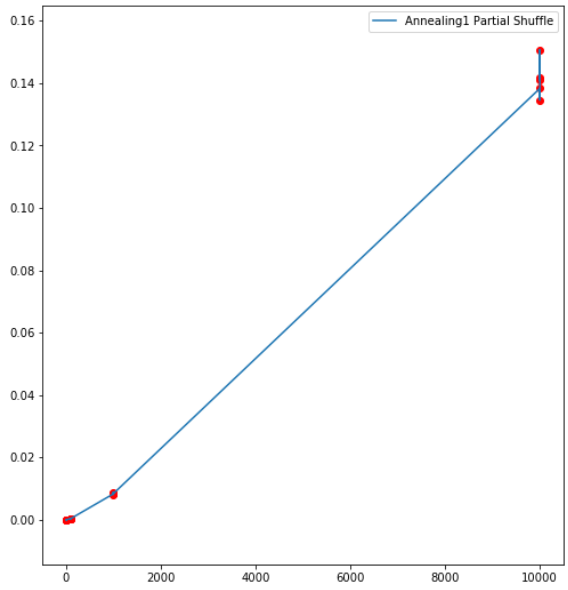
temp1 Sequence = (3^k) / 2 for k >= 1, stop at 3n/4

rep1 Sequence = log2(max Value in temp1Sequence) for all values

In [97]:
```
annealing = np.genfromtxt("data/annealing1.csv", delimiter=',')
annealingpartial = np.genfromtxt("data/partialannealing1.csv",delimiter=',')

plt.subplot(121)
plt.scatter(annealing[:,0], annealing[:,1], c='r')
plt.plot(annealing[:,0], annealing[:,1], linestyle='-', label='Annealing1 Rand
om Shuffle')
plt.legend()

plt.subplot(122)
plt.scatter(annealingpartial[:,0], annealingpartial[:,1], c='r')
plt.plot(annealingpartial[:,0], annealingpartial[:,1], linestyle='-', label='A
nnealing1 Partial Shuffle')

plt.legend()
plt.show()
plt.plot()

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(annealing[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(annealing[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(annealing[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(annealing[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(annealing[i][1])

print("For Random Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(annealingpartial[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(annealingpartial[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(annealingpartial[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(annealingpartial[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(annealingpartial[i][1])

print("For Partial Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
```

```
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))
```





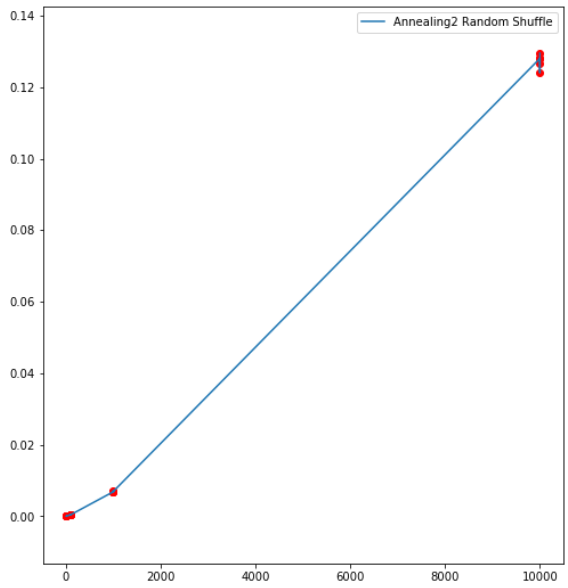

```
For Random Shuffle:
----Average at size 10:  2.86e-05
----Average at size 100:  0.00042259999999999997
----Average at size 1000:  0.0084148
----Average at size 10000:  0.13803400000000002
For Partial Shuffle:
----Average at size 10:  3.62e-05
----Average at size 100:  0.0004078
----Average at size 1000:  0.0084454
----Average at size 10000:  0.14134039999999998
```

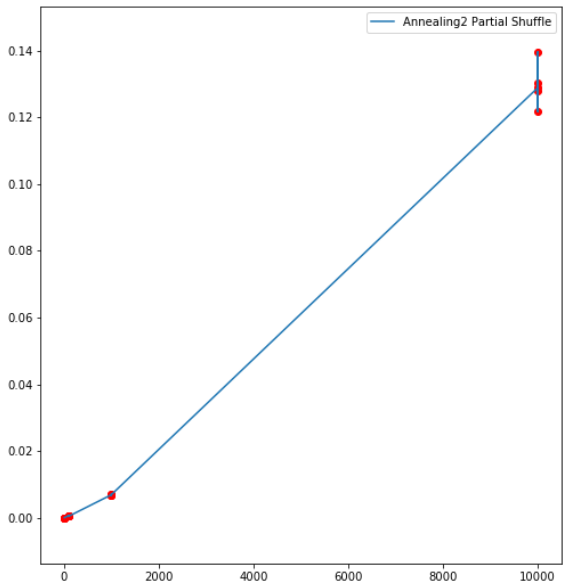
temp2 Sequence = n / (2^k)

rep2 Sequence = size of temp2 Sequence / 2 for all values

In [98]:
```
annealing = np.genfromtxt("data/annealing2.csv", delimiter=',')
annealingpartial = np.genfromtxt("data/partialannealing2.csv",delimiter=',')



plt.subplot(121)
plt.scatter(annealing[:,0], annealing[:,1], c='r')
plt.plot(annealing[:,0], annealing[:,1], linestyle='-', label='Annealing2 Rand
om Shuffle')
plt.legend()

plt.subplot(122)
plt.scatter(annealingpartial[:,0], annealingpartial[:,1], c='r')
plt.plot(annealingpartial[:,0], annealingpartial[:,1], linestyle='-', label='A
nnealing2 Partial Shuffle')

plt.legend()
plt.show()
plt.plot()

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(annealing[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(annealing[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(annealing[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(annealing[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(annealing[i][1])

print("For Random Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(annealingpartial[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(annealingpartial[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(annealingpartial[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(annealingpartial[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(annealingpartial[i][1])
```

```
print("For Partial Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))
```



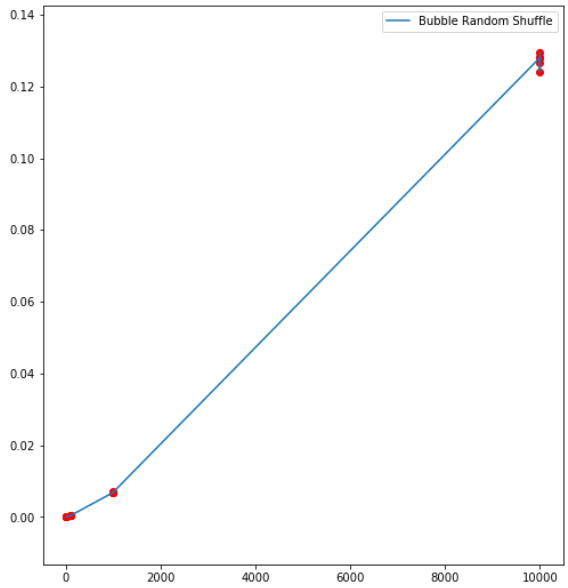


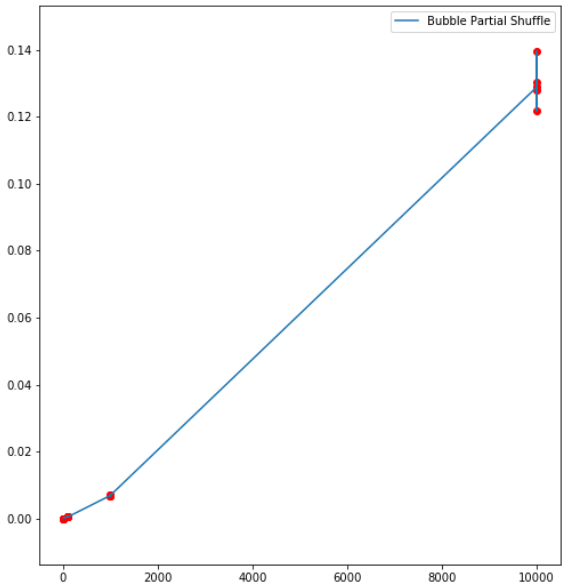

```
For Random Shuffle:
----Average at size 10:  3.26e-05
----Average at size 100:  0.00042259999999999997
----Average at size 1000:  0.0069184
----Average at size 10000:  0.127296
For Partial Shuffle:
----Average at size 10:  2.94e-05
----Average at size 100:  0.0004706
----Average at size 1000:  0.006898799999999999
----Average at size 10000:  0.12966760000000002
```

2. Bubble Sort

In [99]:
```
bubble = np.genfromtxt("data/bubble.csv", delimiter=',')
bubblePartial = np.genfromtxt("data/partialbubble.csv",delimiter=',')


plt.subplot(121)
plt.scatter(annealing[:,0], annealing[:,1], c='r')
plt.plot(annealing[:,0], annealing[:,1], linestyle='-', label='Bubble Random S
huffle')
plt.legend()

plt.subplot(122)
plt.scatter(annealingpartial[:,0], annealingpartial[:,1], c='r')
plt.plot(annealingpartial[:,0], annealingpartial[:,1], linestyle='-', label='B
ubble Partial Shuffle')

plt.legend()
plt.show()
plt.plot()

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(bubble[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(bubble[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(bubble[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(bubble[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(bubble[i][1])

print("For Random Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(bubblePartial[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(bubblePartial[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(bubblePartial[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(bubblePartial[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(bubblePartial[i][1])

print("For Partial Shuffle: ")
```

```
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))
```



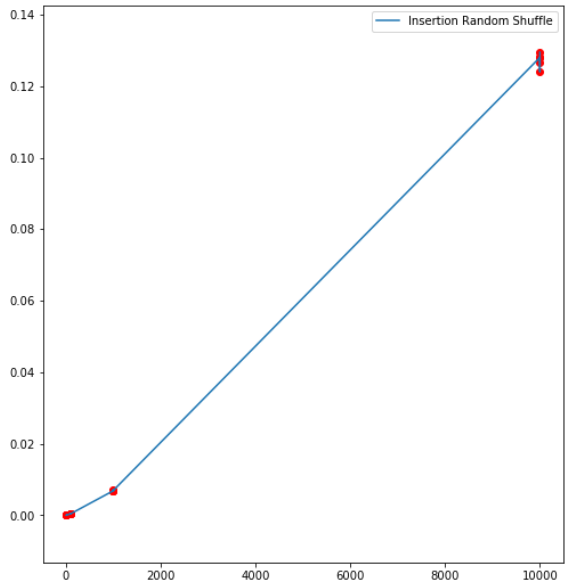


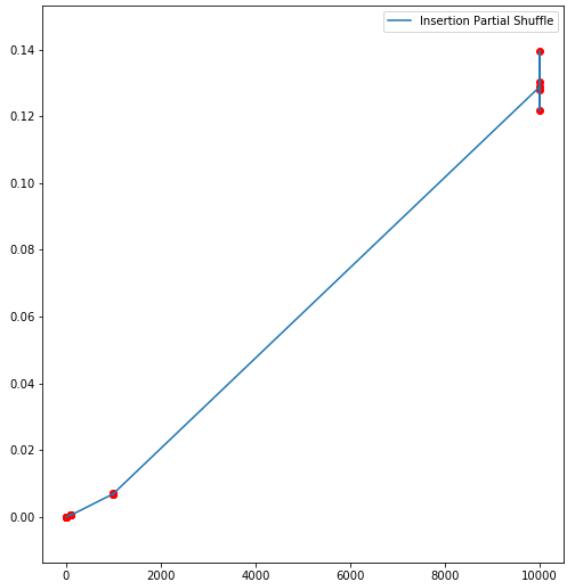

```
For Random Shuffle:
----Average at size 10:  2.4e-06
----Average at size 100:  0.0001702
----Average at size 1000:  0.016827200000000004
----Average at size 10000:  1.7366599999999999
For Partial Shuffle:
----Average at size 10:  2.2e-06
----Average at size 100:  0.0001482
----Average at size 1000:  0.0126788
----Average at size 10000:  1.3324060000000002
```

3. Insertion Sort

In [101]:
```
insertion = np.genfromtxt("data/insertion.csv", delimiter=',')
insertionpartial = np.genfromtxt("data/partialinsertion.csv",delimiter=',')


plt.subplot(121)
plt.scatter(annealing[:,0], annealing[:,1], c='r')
plt.plot(annealing[:,0], annealing[:,1], linestyle='-', label='Insertion Rando
m Shuffle')
plt.legend()

plt.subplot(122)
plt.scatter(annealingpartial[:,0], annealingpartial[:,1], c='r')
plt.plot(annealingpartial[:,0], annealingpartial[:,1], linestyle='-', label='I
nsertion Partial Shuffle')

plt.legend()
plt.show()

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(insertion[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(insertion[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(insertion[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(insertion[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(insertion[i][1])

print("For Random Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(insertionpartial[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(insertionpartial[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(insertionpartial[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(insertionpartial[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(insertionpartial[i][1])

print("For Partial Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
```

```
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))
```





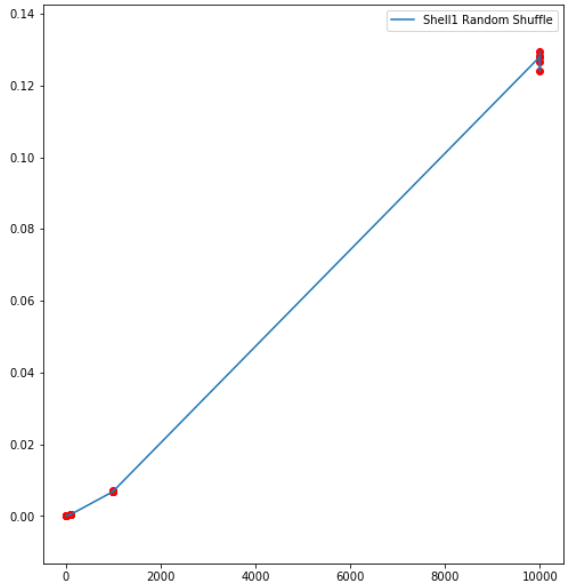

```
For Random Shuffle:
----Average at size 10:  1.2e-06
----Average at size 100:  3.42e-05
----Average at size 1000:  0.0032462000000000003
----Average at size 10000:  0.34067620000000004
For Partial Shuffle:
----Average at size 10:  1e-06
----Average at size 100:  1.2399999999999998e-05
----Average at size 1000:  0.0002212
----Average at size 10000:  0.0029894
```

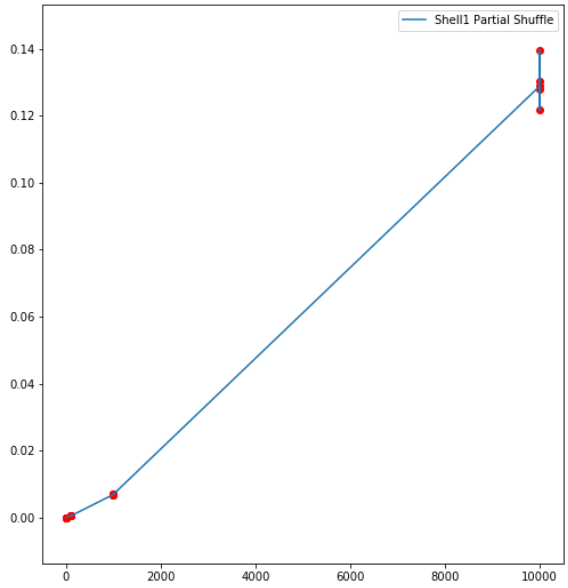
4. Shell Sort

gap1 Sequence = (2^k)n / (3^k) until 0

In [102]:
```
shell = np.genfromtxt("data/shell1.csv", delimiter=',')
shellpartial = np.genfromtxt("data/partialshell1.csv",delimiter=',')


plt.subplot(121)
plt.scatter(annealing[:,0], annealing[:,1], c='r')
plt.plot(annealing[:,0], annealing[:,1], linestyle='-', label='Shell1 Random S
huffle')
plt.legend()

plt.subplot(122)
plt.scatter(annealingpartial[:,0], annealingpartial[:,1], c='r')
plt.plot(annealingpartial[:,0], annealingpartial[:,1], linestyle='-', label='S
hell1 Partial Shuffle')

plt.legend()
plt.show()
plt.plot()

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(shell[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(shell[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(shell[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(shell[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(shell[i][1])

print("For Random Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(shellpartial[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(shellpartial[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(shellpartial[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(shellpartial[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(shellpartial[i][1])

print("For Partial Shuffle: ")
```

```
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))
```



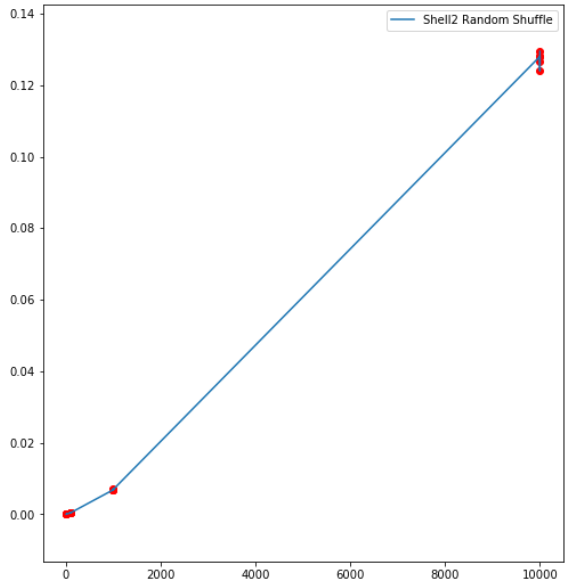


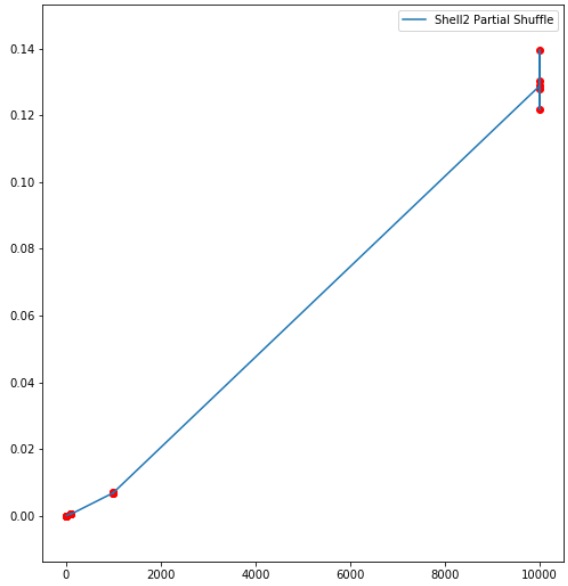

```
For Random Shuffle:
----Average at size 10:  1.6e-06
----Average at size 100:  4.98e-05
----Average at size 1000:  0.0035646
----Average at size 10000:  0.3368624
For Partial Shuffle:
----Average at size 10:  1.6e-06
----Average at size 100:  2.8000000000000003e-05
----Average at size 1000:  0.00047460000000000004
----Average at size 10000:  0.0069592000000000005
```

gap2 Sequence = kN / (3^k) until 0

In [103]:
```
shell = np.genfromtxt("data/shell2.csv", delimiter=',')
shellpartial = np.genfromtxt("data/partialshell2.csv",delimiter=',')


plt.subplot(121)
plt.scatter(annealing[:,0], annealing[:,1], c='r')
plt.plot(annealing[:,0], annealing[:,1], linestyle='-', label='Shell2 Random S
huffle')
plt.legend()

plt.subplot(122)
plt.scatter(annealingpartial[:,0], annealingpartial[:,1], c='r')
plt.plot(annealingpartial[:,0], annealingpartial[:,1], linestyle='-', label='S
hell2 Partial Shuffle')

plt.legend()
plt.show()
plt.plot()

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(shell[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(shell[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(shell[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(shell[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(shell[i][1])

print("For Random Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(shellpartial[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(shellpartial[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(shellpartial[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(shellpartial[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(shellpartial[i][1])

print("For Partial Shuffle: ")
```

```
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))
```



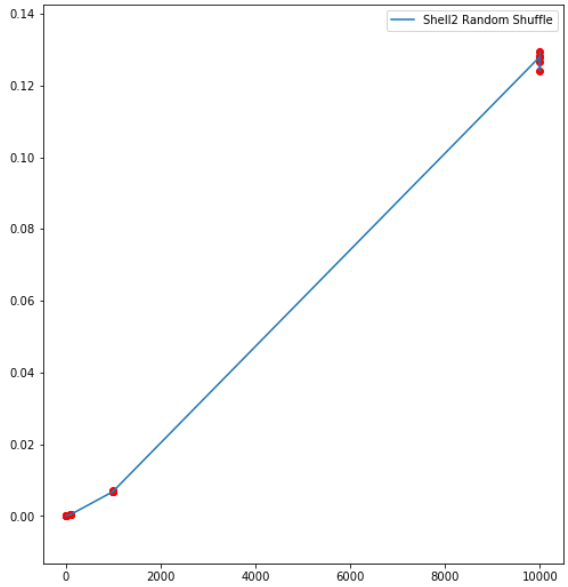


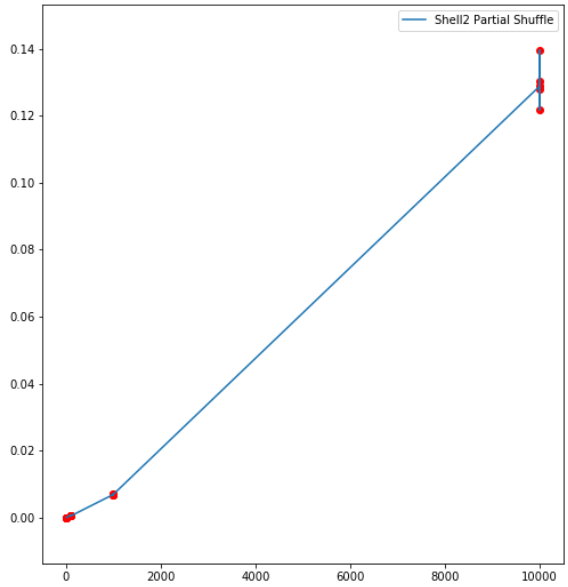

```
For Random Shuffle:
----Average at size 10:  0.002023
----Average at size 100:  4.02e-05
----Average at size 1000:  0.0034583999999999995
----Average at size 10000:  0.3416458
For Partial Shuffle:
----Average at size 10:  1e-06
----Average at size 100:  0.009115000000000002
----Average at size 1000:  0.0003246
----Average at size 10000:  0.0046289999999999994
```

5. Spin The Bottle Sort

In [104]:
```
shell = np.genfromtxt("data/spin.csv", delimiter=',')
shellpartial = np.genfromtxt("data/partialspin.csv",delimiter=',')


plt.subplot(121)
plt.scatter(annealing[:,0], annealing[:,1], c='r')
plt.plot(annealing[:,0], annealing[:,1], linestyle='-', label='Shell2 Random S
huffle')
plt.legend()

plt.subplot(122)
plt.scatter(annealingpartial[:,0], annealingpartial[:,1], c='r')
plt.plot(annealingpartial[:,0], annealingpartial[:,1], linestyle='-', label='S
hell2 Partial Shuffle')

plt.legend()
plt.show()
plt.plot()

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(shell[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(shell[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(shell[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(shell[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(shell[i][1])

print("For Random Shuffle: ")
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))

meanArray1 = []
meanArray2 = []
meanArray3 = []
meanArray4 = []

for i in range(shellpartial[:,0].size):
    if (annealing[i][0] == 10):
        meanArray1.append(shellpartial[i][1])
    if (annealing[i][0] == 100):
        meanArray2.append(shellpartial[i][1])
    if (annealing[i][0] == 1000):
        meanArray3.append(shellpartial[i][1])
    if (annealing[i][0] == 10000):
        meanArray4.append(shellpartial[i][1])

print("For Partial Shuffle: ")
```

```
print("----Average at size 10: " , np.mean(meanArray1))
print("----Average at size 100: " , np.mean(meanArray2))
print("----Average at size 1000: " , np.mean(meanArray3))
print("----Average at size 10000: " , np.mean(meanArray4))
```






```
For Random Shuffle:
----Average at size 10:  2.6000000000000002e-05
----Average at size 100:  0.0017944000000000002
----Average at size 1000:  0.22003500000000004
----Average at size 10000:  32.31712
For Partial Shuffle:
----Average at size 10:  2.16e-05
----Average at size 100:  0.0020158
----Average at size 1000:  0.21825619999999998
----Average at size 10000:  28.87154
```

Conclusion

The first Shell Sort Gap Sequence inspiration came from the example gap sequence from the lecture notes, only changing the ratio at which the values decay from 1/2 to 2/3. The second Shell Sort Gap Seqeunce inspiration came from trying to increase the rate at which the values decay, but not too fast (hence the k product with N) since smaller seqeuences didn't seem to do well.

The first Annealing Sort Temp-Rep Sequences came from an example of a Shell Sort Gap Sequence with a constant Rep Sequence trying to mimic the lecture example of the multi-phase example. I wanted to see if the Shell Sort sequence formula would work with Annealing Sort as well. Constant Rep Sequences also seemed to fully sort the array more often. The second Annealing Sort Temp-Rep Sequence was one of the first I tested taking inspiration from the Shell Sort Sequence example from the lecture. It worked decently so I kept it. The Rep Sequence is also constant but is based off of how many numbers are in the temp sequence. Again, the constant Rep Sequence seemed to make it fully sort the array more often than when I tried to make a more dynamic sequence.

Insertion Sort works well when an array is almost sorted, or close to being sorted. The temps-reps sequences that I used for Annealing Sort worked well at larger arrays. However, Annealing Sort is not 100% guaranteed to work which can cause issues when it doesn't. Shell Sort for both gap sequences seemed to work on par with Insertion Sort for completely random shuffles. Bubble Sort did not do well compared to the other three aleady mentioned, especially when it came to larger arrays. Spin the Bottle Sort wdeteriorated very quickly at larger arrays because of the random nature of the algorithm.