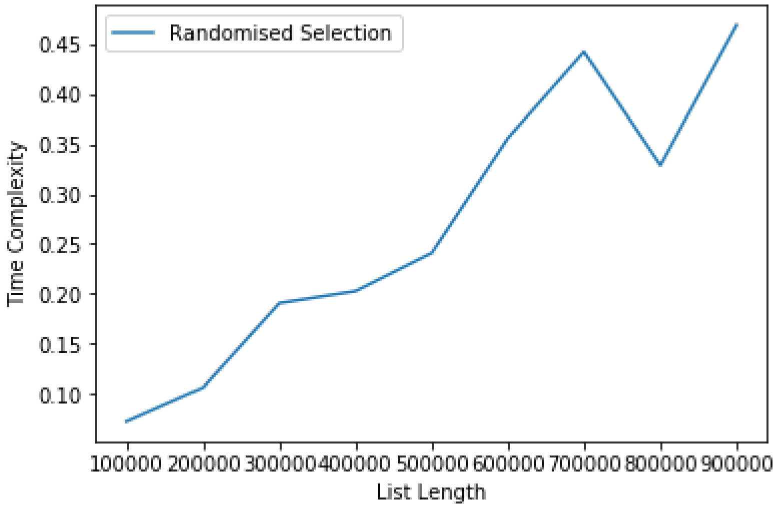
In [21]:

```
import random
import numpy as np

def kthSmallest(arr, l, r, k):

    if (k > 0 and k <= r - l + 1):

        pos = randomPartition(arr, l, r)

        if (pos - l == k - 1):
            return arr[pos]
        if (pos - l > k - 1):
            return kthSmallest(arr, l, pos - 1, k)

        return kthSmallest(arr, pos + 1, r, k - pos + l - 1)

    return 999999999999

def partition(arr, l, r):
    x = arr[r]
    i = l
    for j in range(l, r):
        if (arr[j] <= x):
            arr[i],arr[j]=arr[j],arr[i]
            i += 1
    arr[i],arr[r]=arr[r],arr[i]
    return i

def randomPartition(arr, l, r):
    n = r - l + 1
    pivot = int(random.random() * n)
    arr[l + pivot],arr[r]=arr[r],arr[l + pivot]
    return partition(arr, l, r)


arr = np.random.randint(1,1000,10)
n = len(arr)
print "array generated is ", arr
print "Sorted array looks like ", sorted(arr)
k = int(input("Enter the value of k "))
print "K'th smallest element is", kthSmallest(arr, 0, n - 1, k)
```

```
array generated is  [765 923 787  92 140 996 533  90 215  10]
Sorted array looks like  [10, 90, 92, 140, 215, 533, 765, 787, 923,
996]
Enter the value of k 5
K'th smallest element is 215
```

In [24]:

```
import time
from numpy.random import seed
from numpy.random import randint
import matplotlib.pyplot as plt
import random
import numpy as np

def kthSmallest(arr, l, r, k):

    if (k > 0 and k <= r - l + 1):

        pos = randomPartition(arr, l, r)

        if (pos - l == k - 1):
            return arr[pos]
        if (pos - l > k - 1):
            return kthSmallest(arr, l, pos - 1, k)

        return kthSmallest(arr, pos + 1, r, k - pos + l - 1)

    return 999999999999



def partition(arr, l, r):
    x = arr[r]
    i = l
    for j in range(l, r):
        if (arr[j] <= x):
            arr[i],arr[j]=arr[j],arr[i]
            i += 1
    arr[i],arr[r]=arr[r],arr[i]
    return i

def randomPartition(arr, l, r):
    n = r - l + 1
    pivot = int(random.random() * n)
    arr[l + pivot],arr[r]=arr[r],arr[l + pivot]
    return partition(arr, l, r)




elements = list()
times = list()
for i in range(1, 10):

    a = randint(0, 100000 * i, 100000 * i)
    start = time.clock()
    n = len(a)
    k = n-50
    kthSmallest(a, 0, n - 1, k)
    end = time.clock()
    elements.append(len(a))
    times.append(end-start)



plt.xlabel('List Length')
plt.ylabel('Time Complexity')
plt.plot(elements, times, label ='Randomised Selection')
plt.legend()
plt.show()
```



In [ ]: