6.00 Quiz 3 ----------
Name

1. _____/15
2. _____/15
3. _____/15
4. _____/10
5. _____/10
6. _____/10
7. _____/15
8. _____/5
9. _____/5

---
Athena User Name

---
Recitation hour

Total _____ /100

This quiz is open book and open notes, but do not use a computer (or cell phone!). You have 120 minutes.

Please **write your name on the top of each page**, and your user name and the hour of the recitation you attend on the first page. Answer all questions in the boxes provided.

1) Are each of the following True or False? (15 points)

| |
|---|
| T |

1.1. The result of agglomerative hierarchical clustering depends upon the linkage criterion used.

| |
|---|
| T |

1.2. K means clustering is usually faster than agglomerative hierarchical clustering.

| |
|---|
| F |

1.3. When run on a set of data, the result of k-means clustering does **not** depend on the initial centroids.

| |
|---|
| T |

1.4. Agglomerative hierarchical clustering is a deterministic algorithm.

| |
|---|
| F |

1.5. The continuous knapsack problem **cannot** be solved in O(n log n) time.

2) Consider the following code.

```
yVals = []
for i in range(10000):
    yVals.append(random.gauss(0, 4))
xVals = pylab.arange(10000)
a, b, c = pylab.polyfit(xVals, yVals, 2)
print round(a)
print round(b)
print round(c)
pylab.plot(sorted(yVals, reverse = True))
pylab.xlim(0, 10000)
pylab.ylim(-20, 20)
```

2.1. What does it print? (8 points)

```
0.0
0.0
0.0
(each 0.0 may be -0.0, but don't care too much about the - sign)
```

2.2. Draw an approximation to the plot it is likely to produce (7 points)

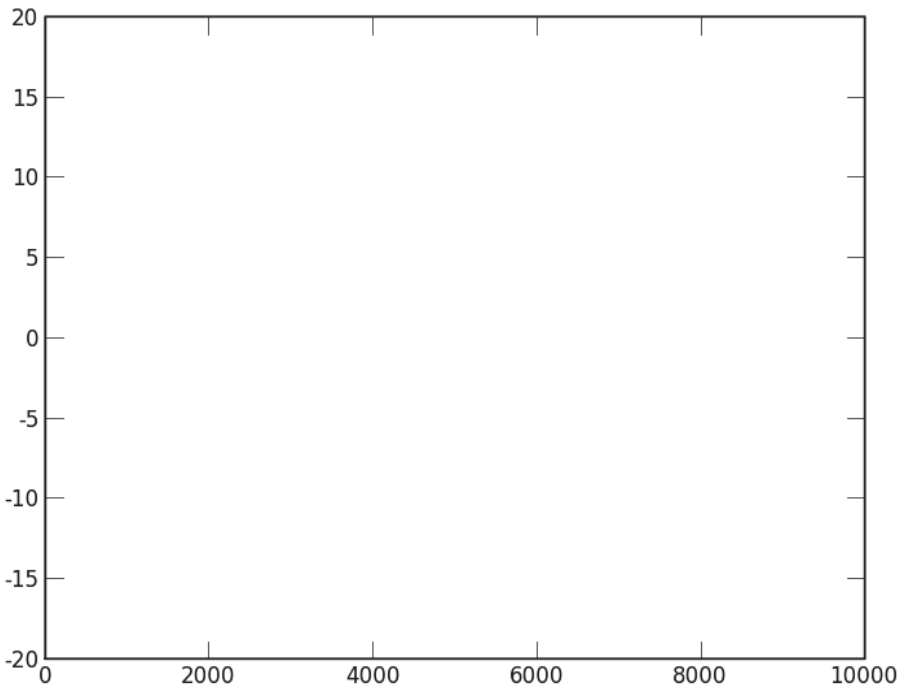
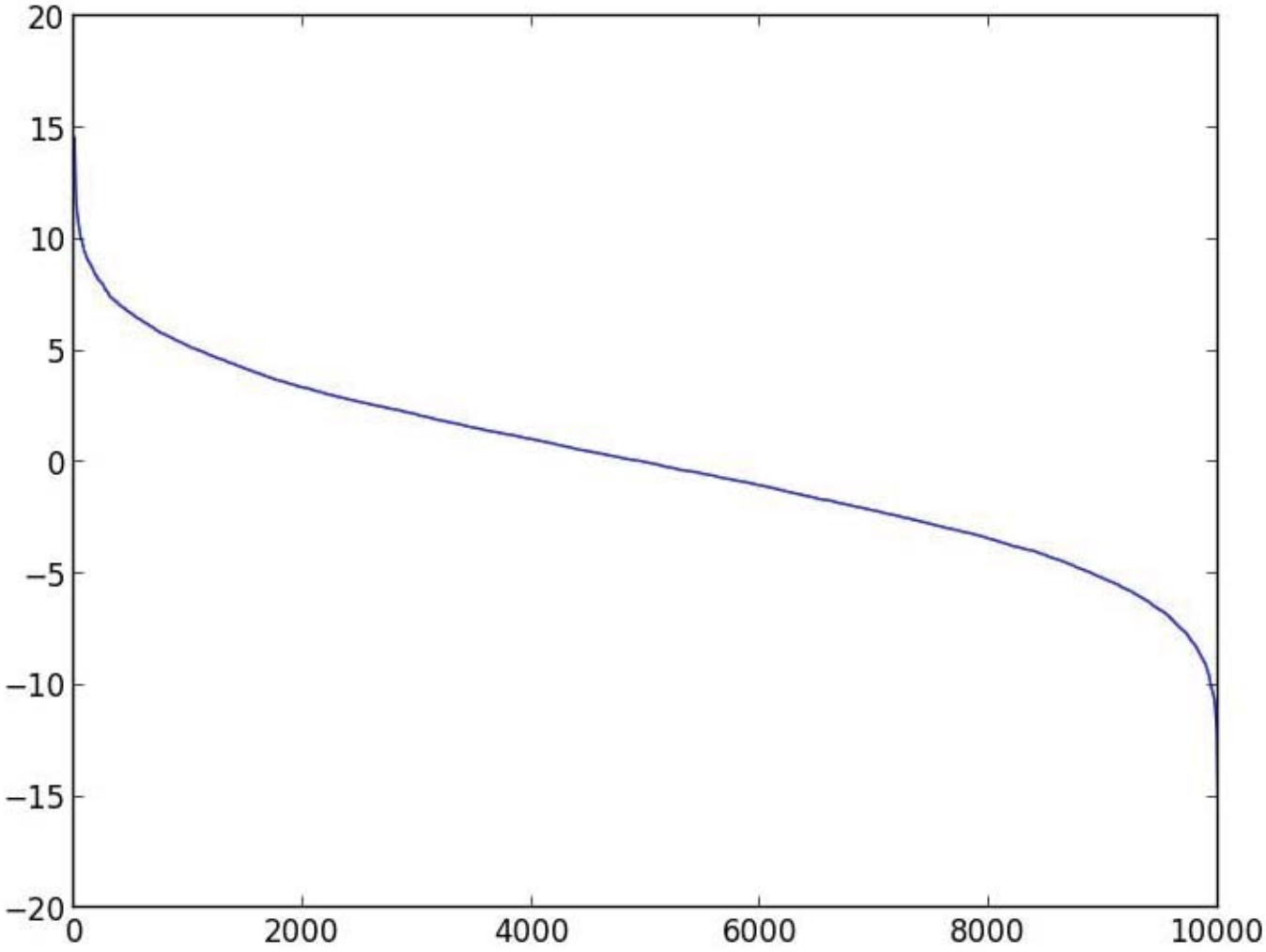


figure for problem 2.2



3) 1000 students took an online course. ¼ of them were from Africa, ¼ from Europe , ¼ from South America, and ¼ from Asia. At the end of the course, the instructor observed that of the top 100 grades, 35 belonged to students from one geographical area (South America). He argued that since the expected number of students from each area in the top 100 was 25, this was unlikely to have happened by pure chance. Write a program that returns an estimate of the probability of this happening purely by chance. (15 points)

```
Abstract the problem: calculate the probability that out of the top 100
students, at least 35 were from one region (from the same one region,
but not tied to South America, any arbitrary region of the 4).


import random

N = 1000      # number of students
cutoff = 100  # the top 100 cutoff
M = 35        # 35 or more were from the same one region

Nsims = 100000  # total number of simulations to do

def simulate():
    """
    assume 0 stands for students from Africa, 1 for Europe,
    2 for Europe, 3 for Asia.
    Randomly shuffle them to get the sorted list.
    returns whether in the top 100 students, 35 or more were from 1 region.
    """

    students = [0]*(N/4)+[1]*(N/4)+[2]*(N/4)+[3]*(N/4)
    random.shuffle(students)
    total = [0]*4
    for i in xrange(cutoff):
        total[students[i]] += 1
    return max(total) >= M

def monteCarlo():
    result = 0
    for i in xrange(Nsims):
        if simulate():
            result += 1
    return result / float(Nsims)

(If you run it, the resulting probability will be around 0.048)
```