**Final Exam Problem 9 - Your program timed out.**

[IanGlencross](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/511888) 2 days ago

When I run my code with 1000 students I get a timeout message: Your Program timed out. Check for very slow code or infinite loops.

But with 500 students i get very close to the 0.51666 result. (in fact any number over 200 seems to give an ok result).

Is there some way I can improve this code to get it to run with 1000 students <https://gist.github.com/4547490#file-atleast30>

1. [0](javascript:void(0)) [bluefin](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/66101)

2 days ago

Just run it with 250 students and count the random numbers are over .9. Compute probability p of less than 30 over trials and return 1.-p\*\*4.

Top of Form



Bottom of Form

1. [0](javascript:void(0)) [dnc](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/23266)

2 days ago

You don't need to generate any actual grades for this problem. For each trial, you only need to randomly pick one of the four areas 100 times. If any area gets picked more than 30 times, increment a counter.

After numtrials, return counter/numtrials

I see a few people had a time out on this problem, it's a shame because it seems like your code does solve the problem asked and although it generates a bunch of detail not required for this exact problem, it would be much simpler to adapt to ask other questions of the data.

* + Check out the Statistics Lecture code presented regarding 30% likelihood of anorexics born in June. This code can easily be adapted from that

–posted 2 days ago by [PBK](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/428591)

* + I agree with dnc. You don't need grades, you need class rankings.

I just generated an ordered list of 1000 students and shuffled them repeatedly. After each random.shuffle(list), I dealt them into four hands/regions, and sorted them in order. If the 30th element (i=29) is less than 100, then that group had more than 30 of the top students. Increment the count and shuffle on!

def test(numTrials):

"""

Uses simulation to compute and return an estimate of

the probability of at least 30 of the top 100 grades

coming from a single geographical area purely by chance

"""

numStud = 1000

quart = numStud/4

t = range(numStud)

count = 0

for n in range(numTrials):

random.shuffle(t)

s1 = sorted(t[0:quart-1])

s2 = sorted(t[quart:2\*quart-1])

s3 = sorted(t[2\*quart:3\*quart-1])

s4 = sorted(t[3\*quart:numStud-1])

if (s1[29]&lt;100) or (s2[29]&lt;100) or (s3[29]&lt;100) or (s4[29]&lt;100):

count += 1

return float(count)/numTrials

–posted 2 days ago by [GraceLAX](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/174846)

* + You don't need to go this far either. The question was "Write a function that uses simulation to compute and return an estimate of the probability of at least 30 of the top 100 grades coming from a single geographical area purely by chance"

The important information for the question is which students do we care about? - only the top 100. Where do they come from? - equal probability of 4 areas. I posted my solution in this thread -

<https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/i4x-MITx-6_00x-course-2012_Fall/threads/50f632f3f1fdab2600000026>

–posted 2 days ago by [dnc](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/23266)

* + @dnc: what's a simulation without grades? :)

–posted 2 days ago by [Coen](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/671102)

* + I see now that my approach is wrong but not for the reason you suggest Coen. See pyTony and IvanaKs posts to see why.

A simulation without grades is exactly that, a simulation in which you have abstracted out detail which is not needed to answer the question. You don't need to simulate the entire cosmos to find the probability of a fair dice roll.

–posted 2 days ago by [dnc](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/23266)

* + hi m8. I wasn't suggesting your calculation was wrong or right. I was just pointing out it wasn't a proper simulation. :) anyways mine seems to work gr8 and has less lines of code... grades and all

–posted 2 days ago by [Coen](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/671102)

* + I was meaning that without grades, it still is a proper simulation, just a more abstract one. You can simulate the geography of the top 100 students without including the actual grades in the simulation.

–posted a day ago by [dnc](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/23266)

Top of Form



Bottom of Form

1. [0](javascript:void(0)) [Mattchoo](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/382433)

2 days ago

One thing that might cause a problem is your implementation. I tried using a class for students but it timed out on me as well. I switched to using 4 lists and 250 grades for each list and finding the top 100 out of the 1000 total and that did work. Hope you figure something out.

Top of Form



Bottom of Form

1. [0](javascript:void(0)) [PBK](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/428591)

2 days ago

Check out the Statistics Lecture code presented regarding 30% likelihood of anorexics born in June. This code can easily be adapted from that.

Top of Form



Bottom of Form

1. [0](javascript:void(0)) [pyTony](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/311172)

2 days ago

Look also thread <https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/i4x-MITx-6_00x-course-2012_Fall/threads/50f632f3f1fdab2600000026>

I run through my thinking on one post and link to code for my solution.

Top of Form



Bottom of Form

1. [1](javascript:void(0)) [FlyingDiver](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/104291)

2 days ago

You guys are all way over thinking this.

import random

def test(numTrials):

students = 100

over = 0

for i in range(numTrials):

count = [0,0,0,0]

for i in range(students):

region = int(random.random() \* 4)

count[region] += 1

if max(count) &gt;= students \* 0.3:

over += 1

return float(over) / float(numTrials)

* + I get probability over 55 %, so this is oversimplified. I trust more mine less than 52 % (the checker says *of the calculated probability of 0.5166.*)

–posted 2 days ago by [pyTony](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/311172)

Top of Form



Bottom of Form

1. [0](javascript:void(0)) [IvanaK](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/113391)

2 days ago

The result is close enough for the grader for this particular set of numbers, but the probability of picking the next student from one particular country is not always 1/4. It changes after every draw. If you picked already 29 students from one country out of 1000 students, the probability of picking the 30th from the same country is 221/971 (.227). If you say it is close enough to 1/4 you can use probability 1/4 every time you are drawing next student. Obviously, the grader accepted this. The difference of results is on the order of ones of percent. If you are choosing 100 out of 10000, the difference is really small. But if you are choosing 100 out of 200 the difference is significant. If you take into account that drawings are dependent then result is around 0.52.

Top of Form



Bottom of Form

1. [0](javascript:void(0)) [Coen](https://www.edx.org/courses/MITx/6.00x/2012_Fall/discussion/forum/users/671102)

2 days ago

Hi folks.

Correct me if i'm wrong, but the question was:

Write a function that uses **simulation**

I agree you dont need grades it to pass the grader, but it doesn't feel like a proper simulation without them.

Ok, this is how i did it:

- represent a grade as [grade, region]

- generate 250 for each region and add them to 1 list

- sort the list

- splice off the top 100 records

- now count how many grades each region has

Done :)

Top of Form



Bottom of Form

Top of Form

**Post a response:**

Bottom of Form