Mark Eatough

CSIS 2430 9:00 Class

Programming Project 9

Bacterial Growth Program

Assignment objective:

A deadly bacterium is extremely active and will kill a person if they have more than 1 million bacteria in their body at any given time. There is a single cure that takes 2 hours to begin working to kill the bacteria. If a person starts out with 2 bacteria and doubles every 20 minutes, how long before they are dead? When do you need to start the cure so the patient does NOT die?

What Worked?:

I created a time object to keep track of how much time has passed in terms of hours and minutes. Then I split the problem into smaller parts. In one method I calculate the amount of bacteria present in the patient in terms of intervals n. In another method I call the first method, incrementing n as we go, and my time object by 20 each iteration. To get a more precise time once the bacteria goes over 1 million I get the amount of bacteria at n-1, calculate the rate of increase, and then figure out how many minutes it will take to reach 1 million

What did not work?:

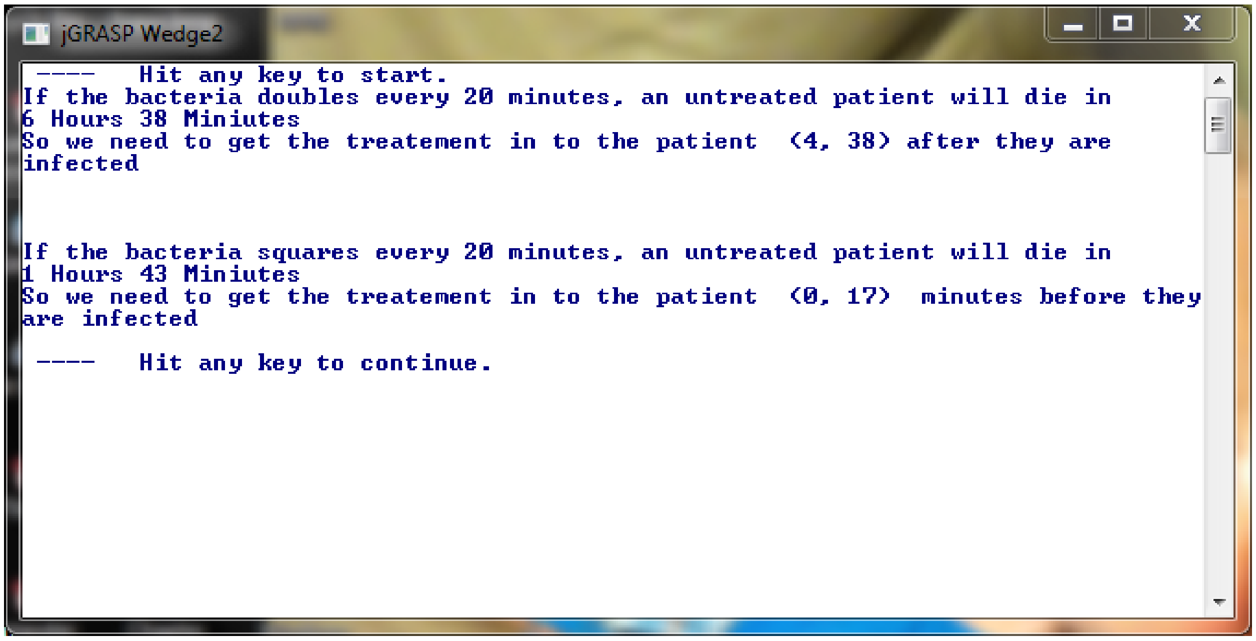
The equation we were given in one of the videos was that bacteria grew at a rate of $2*2^n$, in which doubles the bacteria at each time interval, but in the example on the video the bacteria was squared at each interval. So to be safe I wrote a method to handle each of these outcomes. This was a very easy program to code as you said, so I had no other major issues.

Comments:

This was a very simple problem as you said, but it was still very interesting. I was excited to get to use some simple calculus to be able to find the exact time that the bacteria would reach one million.

```
'''
************************************************************
 * Discrete Structures
 * Bacteria Growth Program
 * Programmer: Mark Eatough
 * Course: CSIS 2430
 * Created October 27, 2013

 ************************************************************
'''

import math

#class to create and manage time object
class Time:
   def __init__(self, hours, miniutes):
      self.hours = hours
     self.miniutes = miniutes
      while(self.miniutes >= 60):
         self.hours += 1
         self.miniutes -= 60
   #toString method equivalent
   def __str__(self):
      return("{0} Hours {1}
Miniutes".format(self.hours,self.miniutes))
   #method to add two times together
   def addTime(self, otherTime):
      myHours = self.hours+otherTime.hours
      myMiniutes = self.miniutes+otherTime.miniutes
      while(myMiniutes >= 60):
         myMiniutes -= 60
         myHours += 1
      myTime = Time(myHours, myMiniutes)
      return myTime

   def subTime(self, otherTime):
      myHours = 0
      otherTime.miniutes += (otherTime.hours*60)
     myMinutes = self.miniutes + (self.hours*60) -
otherTime.miniutes
      while(myMinutes > 60):
         myMinutes -= 60
         myHours += 1
      myTime = (int(math.fabs(myHours)),
int(math.fabs(myMinutes)))
      return myTime
#end time class
```

```
def squareBacteria(n):
   bacteria = 2
   while(n>0):
      bacteria*=bacteria
      n-=1
   return bacteria

def doubleBacteria(n):
   bacteria = 2
   while(n>0):
      bacteria*=2
      n-=1
   return bacteria

def doubleToMillion(timeToDouble):
   minsToDouble = Time(0,timeToDouble)
   minsPassed = Time(0,0)
   bacteria = 0
   n = 0
   while(bacteria<1000000):
     bacteria = doubleBacteria(n)
      if(bacteria > 1000000):
         bacteria= doubleBacteria(n-1)
         bacteriaToGo = 1000000-bacteria
         increasePerMin = bacteria/20
         timeLeft = bacteriaToGo/increasePerMin
         timeAdded = Time(0,timeLeft)
         minsPassed = Time.addTime(minsPassed, timeAdded)
         bacteria = 1000000
      else:
         minsPassed = Time.addTime(minsPassed, minsToDouble)
         n+=1
   return minsPassed

def squareToMillion(timeToDouble):
   minsToDouble = Time(0,timeToDouble)
   minsPassed = Time(0,0)
   bacteria = 0
   n = 0
   while(bacteria<1000000):
      bacteria = squareBacteria(n)
      if(bacteria > 1000000):
         bacteria = squareBacteria(n-1)
         bacteriaToGo = 1000000-bacteria
         while(bacteria < 1000000):
            increasePerMin = bacteria*2
            bacteria+=increasePerMin
            minsPassed = Time.addTime(minsPassed, Time(0,1))
      else:
         minsPassed = Time.addTime(minsPassed, minsToDouble)
         n+=1
   return minsPassed



double = doubleToMillion(20)
diff1 = Time.subTime(double, Time(2,0))
print "If the bacteria doubles every 20 minutes, an untreated
patient will die in ", double
print "So we need to get the treatement in to the patient ",
diff1, "after they are infected"

square = squareToMillion(20)
diff2 = Time.subTime(square, Time(2,0))
print "\n\n\nIf the bacteria squares every 20 minutes, an
untreated patient will die in ", square
print "So we need to get the treatement in to the patient ",
diff2, " minutes before they are infected"
```



```
 ----   Hit any key to start.
If the bacteria doubles every 20 minutes, an untreated patient will die in
6 Hours 38 Miniutes
So we need to get the treatement in to the patient  (4, 38) after they are
infected



If the bacteria squares every 20 minutes, an untreated patient will die in
1 Hours 43 Miniutes
So we need to get the treatement in to the patient  (0, 17)  minutes before they
are infected

 ----   Hit any key to continue.
```