PROBLEM OF THE MONTH

POM

2016-2017

December, 2016

MATHEMATICS

5 points:

A sequence of numbers is defined by the following recurrent relation $a_{n+1} = \frac{1}{a_n + 4}$, with $a_1 = 1$. Find first few members of the sequence (you can use computer if you wish). What do you notice? Can you guess or derive a_{2017} with good accuracy?

10 points:

A sequence of numbers is defined by the following recurrent relation $a_{n+1} = \frac{n+1}{n} a_n - \frac{2(2n+1)}{n^2(n+1)}$, with $a_1 = 3$.

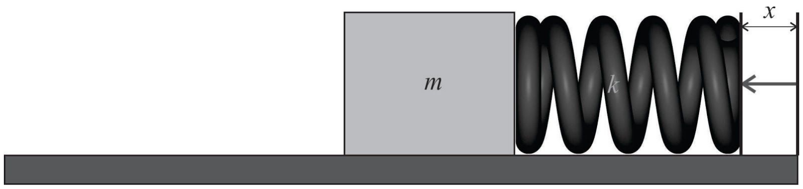
Compute a_{2017} with the accuracy of at least 10^{-100}.

PHYSICS

5 points:

An aluminum sphere of volume V= 200 ml, with a void inside, is floating half-immersed in water. Find the volume of of the void.

10 points:



A block of mass m standing on a flat surface is very slowly pushed through a massless spring with the spring constant k. At first the block stands still and the spring gets compressed, but then at some point when the spring is compressed by distance x the block starts moving. Find the distance that block will travel once it starts moving (assume that the far side of the spring stands still while the block moves). The coefficient of static friction is $\mu_1 = 0.5$, the coefficient of kinetic (sliding) friction is $\mu_2 = 0.4$

.

CHEMISTRY

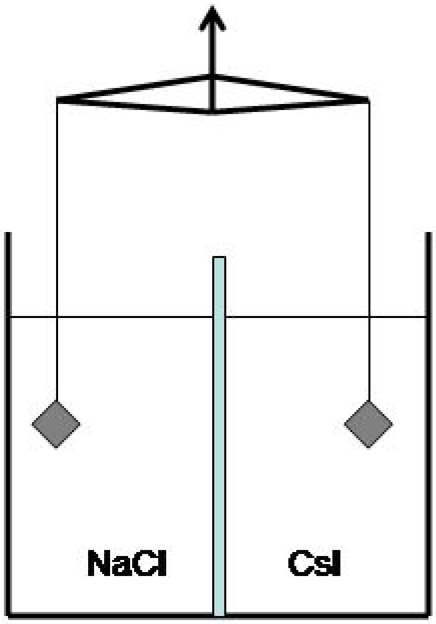
5 points:

In Amazonian selva, many illegal gold miners use a very dangerous and obsolete technology for final step of gold extraction and purification. What this technology consists in, what chemical and physical processes are behind it, and why is this technology so dangerous specifically for selva ecosystem?

There WILL be a hint for this problem.

10 points:

A glass beaker is separated by a semipermeable membrane in two equal parts as shown in the figure. A saturated solution of sodium chloride is in the left part of the beaker, and a saturated solution of cesium iodide is in the right part. Two strings are attached to the ends of the balance as shown on the figure, and the crystals of sodium chloride and cesium iodide are attached to the ends of these strings (see the figure). Since a saturated solution of some substance is the solution that cannot dissolve additional amount of that substance, and because each crystal is immersed into the solution of the same salt (sodium chloride crystal is in the saturated solution of NaCl, and cesium iodide is in CsI), each crystal is in equilibrium with the solution, and the balance is in equilibrium too.



Describe what will happen if we let this system stand for a while.

Two comments: First, ignore the effect of water evaporation (it is too slow in this case). Second, in this experiment, the semipermeable membrane is permeable for water molecules, but not for cesium iodide or sodium chloride.

There WILL be a hint for this problem.

BIOLOGY

5 points:

Most of the major animal groups rely on contraction of muscle for movement, respiration, and blood circulation. Muscle contractions are initiated by the firing of motor neurons that often link up to multiple muscle fibers. As a result, only one or a few neurons are needed to effectively trigger a relatively large muscle, firing each time the muscle needs to contract.

However, nature has a few examples where muscles can contract spontaneously, without the input of the nervous system (not to be confused with voluntary vs. involuntary muscle control by the brain). Can you think of an example of an animal or a type of muscle where contractions are uncoupled from neuronal activity? Why might such “uncontrolled” contractions be of an advantage there?

10 points:

Tomorrow afternoon you get a phone call from the nearest Center for Disease Control informing you that a brand new hospital is being built in your area. As luck would have it, all of the CDC’s own experts have already taken off for the holidays, and you are being asked to help with making sure that the new hospital is built in accordance with safety guidelines for infectious disease control. You call the agency in charge of the hospital’s construction, and find out that plans for the new building include an emergency room, an operating room, a tuberculosis ward, and a number of rooms for generally admitted patients. You need to decide how to set up the ventilation in these different parts of the hospital so as to keep infectious microorganism populations from becoming a danger to the people inside the hospital as well as to all those living in your area. You have three choices to set up the airflow in any given room. These are: 1. outside air to flow into the room, 2. inside air to flow out, or 3. air to enter and leave the room through a system of filters separating it from the rest of the hospital’s airflow.

What would be the proper choice of airflow for the different rooms in the hospital, and why?

Bonus Hint: Microorganisms flourish when they have easy access to nutrients and (importantly) when they see little or no competition from other organisms.

COMPUTER SCIENCE

- You can write and compile your code here: http://www.tutorialspoint.com/codingground.htm
- Your program should be written in Java or Python
- No GUI should be used in your program: eg., easygui in Python. All problems in POM require only text input and output. GUI usage complicates solution validation, for which we are also using *codingground* site. Solutions with GUI will have points deducted or won't receive any points at all.
- Please make sure that the code compiles and runs on http://www.tutorialspoint.com/codingground.htm before submitting it.
- Any input data specified in the problem should be supplied as user input, not hard-coded into the text of the program.
- Submit the problem in a plain text file, such as .txt, .dat, etc.
  No .pdf, .doc, .docx, etc!

Introduction: Maps

We all quite familiar with geographical maps. They depict continents, oceans, etc. on a sheet of paper. The key point here is that they "translate" 3 dimensional surface of Earth into 2 dimensional picture. By the way, this representation is not unique. You may heard about tens of different map projections. Mathematics borrows this meaning and says, for example, that a function is basically a map that gets an input (say, x) and produces an output (say, y) or "maps x to y". This sense of association is also used in computer science. In some computer languages maps are initialized with arrows which stresses this property, for example,

In *Perl* (maps are called hashes there):

%data = (1 => "one", 2 => "two");

In *Scala*:

val data = Map(1 -> "one", 2 -> "two")

Thus, we are talking about 2 things where one is associated with the other. The first thing is called "key" and the second one is called "value", i.e. key => value. So, given a key it's very easy and fast to find its value. The reverse is not true.

If you remove the values from this data structure you essentially end up with Sets that we studied last time.

The keys are usually stored in an array. As you know, you can read any Key from it very fast if you know it's location, index in our case. So, how we translate a Key into its index? That's what a hash function does. In general case it's quite complicated topic, but we can give you a taste of how it can be done. Let's get some number out of Key. If you started with an integer then you already have it. But what if we have a string? We can assign some number to each letter, for example, 'A' can be 1, 'Z' = 26, 'a' = 27, 'z' = 52. Then you sum up all the numbers.

Suppose you create an array of 10 elements. Then you compute the key value modulo 10. For example, for the word “ABC” in our example it’ll be 1+2+3 % 10 = 6 % 10 = 6.

But what happens when your key hash value is the same for different keys? It’s called a hash collision. There are multiple strategies how to deal with this. The simplest one is to store a list of key-value pairs that correspond to this hash value. Obviously this degrades performance. You just jumped with 1 operation (of computing the hash value) into a proper key place but then you have linearly scan this list to find what you’re looking for. Usually this happens when you’ve put a lot of values into a small map, i.e. your hash array is not large enough. The usual remedy in this case is to create a new bigger map and rehash all the content. This will disperse all the key-value pairs into more hash buckets and thus your lists will become smaller (remember that you modulo k, where k is your hash array length). You could have started with a large array, but then your map structure would be sparse, and you would waste a lot of memory. This is a classical trade-off between CPU (number of things you have to compute) and memory.

Similar to Sets, keys in maps are uniquely identify their values, so when you assign

data{1} = “one”

and then

data{1} = “uno”

then your map will have only one pair: 1=>”uno” which will override the previous value. Different languages have different syntax for maps. They all typically allow to add a new key-value pair, delete one, find whether a given key is already in the map, and find its associated value. Look up the maps documentation for the language of your choice to learn how to implement them in your code. (Note: in Python maps are called Dictionaries).

5 points:

You are probably familiar with a simple method to scramble your secret messages where each letter in the original text is replaced with some other letter in “ciphertext”. Your assignment this month will be to write a simple descrambler. Your program should input: original plaintext alphabet and the corresponding ciphertext alphabet, for example like this:

Original: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext: ZYXWVUTSRQPONMLKJIHGFEDCBA

Then your program should input an encrypted message and print a corresponding descrambled one. Your program should be using maps.

10 points:

Problem: Scrabble-Boggle game.

This month your assignment will be to implement a particular variant of the game Boggle. Not sure if you have ever played Boggle, and if not - it’s a lot of fun. In Boggle, you are given 16 letters organized as a 4x4 square. Your task is to find all 3 or more letter long words using the following rule: The letters must be adjoining in a 'chain': letter squares in the chain may be adjacent horizontally, vertically, or diagonally. For example, given this board:

Z	**E**	**G**	Z
L	Z	**O**	Z
Z	Z	Z	Z
X	**O**	**B**	Z

you could find words LEG, LEGO, EGO, GEL and BOX. Note that each letter square can be used only once per word: you cannot make word EGG in our example. To make your program simpler, all the words will have to belong to a given set of allowed words (super-abridged dictionary of English language ς). We suggest not more than 50-100 words. To make the game even more interesting, each found word will have associated numeric value, using letter value system from Scrabble. Specifically, letters from A-Z have the following corresponding values:

A-1,B-3,C-3,D-2,E-1,F-4,G-2,H-4,I-1,J-8,K-5,L-1,M-3,N-1,O-1,P-3,Q-10,R-1,S-1,T-1,U-1,V-4,W-4,X-8,Y-4,Z-10
So, LEG has a value of 4, LEGO - 5, etc.

Your program may have a built-in map of letter values, and we also suggest storing a set of allowed words. Then your assignment is: enter the 4x4 Boggle board, then figure out all possible words that belong to the set of allowed words, calculate their Scrabble value and print them in the order of decreasing value.

For example, with the board from the example above, your program would print:

```
BOX - 12
LEGO - 5
EGO - 4
```

LEG - 4
GEL - 4
(assuming all these 5 words are in allowed words set)