PROBLEM OF THE MONTH

November, 2016

MATHEMATICS

5 points:

Is it possible to arrange all natural numbers from 1 to 42 in a table with 7 columns and 6 rows so that the sum of numbers in each row is the same?

Hint: What is the sum of all numbers from 1 to 42?

Answer: No

Solution: The sum of all numbers from 1 to 42 can be found as a sum of arithmetic progression $(1+42)*42/2 = 43*21$. This sum is not divisible by 6 and, therefore, divide this sum equally between 6 rows is not possible.

10 points:

Numbers from 1 to 100 are written on the blackboard. In one operation, you can erase any two of the numbers, say, (a,b) and replace the erased pair by the number $a+b+ab$. What number will be left on the blackboard after 99 such operations?

Hint: Notice that $a+b+ab=(a+1)(b+1)-1.$

Answer: $101!-1$

Solution: Every operation replaces the pair (a,b) by the number $c=a+b+ab=(a+1)(b+1)-1.$ We notice that $c+1=(a+1)(b+1).$ In other words if we think about numbers shifted by one, the pair $(a+1,b+1)$ is replaced by

their product $(a+1)(b+1)$. After 99 operations the number x left on the blackboard will be given by $x+1=(1+1)(2+1)...(100+1)=101!$. Therefore, $x=101!-1$.

PHYSICS

5 points:

During a rainstorm, an empty bucket moving up with vertical speed V gets filled in 2 minutes. The same bucket moving down with the same vertical speed V gets filled in 8 minutes. How long will it take to fill a motionless bucket?

Hint: . Let the rain drops fall with some unknown speed U. The rate at which the bucket is filled is proportional to that speed. Therefore, the time to fill the bucket is T=k/U, where k is certain constant. Now consider the case when the bucket goes up and down.

Answer: 3.2 min.

Solution: Let the rain drops fall with some unknown speed U. The rate at which the bucket is filled is proportional to that speed. Therefore, the time to fill the bucket is T=k/U, where k is certain constant. Now consider the case when the bucket goes up . In this case, speed of raindrops with respect to the bucket is U+V, ant time to fill it is:

$$\frac{k}{U+V} = 2min\,.$$

When the bucket moves down, the rain drops move with respect to it at speed U-V. Therefore,

$$\frac{k}{U-V} = 8min\,.$$

Now we can find the unknown time T if we note that:

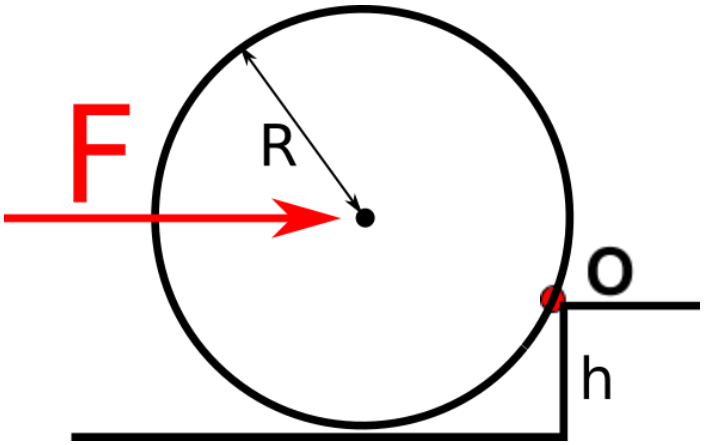
$$\frac{1}{T} = \frac{U}{k} = \frac{U+V}{2k} + \frac{U-V}{2k} = \frac{1}{2\times 2min} + \frac{1}{2\times 8min} = \frac{5}{16min}$$

T=3.2 min

.

10 points:

What horizontal force do you need to apply to this cylinder, mass ***M,*** to make it roll over the edge?



Hint: Imagine that the cylinder had just detached from the floor. Sketch all the forces acting on it. In order for it to roll over, the net torque acting on the cylinder, *with respect to the point O*, should be directed clockwise.

Answer: $\frac{Mg\sqrt{h(2R-h)}}{R-h}$

Solution: Imagine that the cylinder had just detached from the floor. In order for it to roll over, the net torque acting on the cylinder, *with respect to the point O*, should be directed clockwise. There are only two forces that create this torque: F that has lever arm R-h, (it rotates the cylinder clockwise), and weight Mg, with lever arm d (it rotest the cylinders in the opposite direction.

This means that the cylinder will roll over if

$F(R-h)-Mgd>0$

Here $d=\sqrt{R^2-(R-h)^2}$, by Pythagorean Theorem.

Therefore, the lowest force sufficient to make the cylinder to roll over the edge is:

$$F=\frac{Mg\sqrt{R^2-(R-h)^2}}{R-h}=\frac{Mg\sqrt{h(2R-h)}}{R-h}$$

CHEMISTRY

5 points:

You probably know that some dyes can change color in acidic or basic media. For example, methyl orange is red in an acidic media (for example, in aqueous acetic acid), becomes yellow in a basic media (for example, in sodium hydroxide solution), and becomes red again if an excess of some acid is added to this yellow basic solution. Such dyes are called *indicators*. Not only they change color in the presence of acids or bases, they do that reversibly. Try to experiment in your kitchen to identify substances that behave like indicators. Of course, it is hardly possible to find a dye that undergo such dramatic color change as methyl orange does, but there should be some materials or food in your kitchen that demonstrate an indicator like behavior.

Do experiments, take pictures showing color changes and send us your pictures with the description of the experiment.

For your experiment, you will need some acid and some base. You can use a lemon juice or vinegar as an acid, but it is not easy to find a reasonably strong base in your kitchen. You can prepare it by decomposing baking soda. Take a teaspoon of baking soda, put it into a cup, add some water (about ⅓ of cup's volume) and microwave for about 20-30 seconds (until the liquid started to boil). That will lead to decomposition of the baking soda according to the equation:

$2\ NaHCO_3 \rightarrow Na_2CO_3 + H_2O + CO_2$

The product of this reaction is washing soda (Na_2CO_3), and it makes a solution strongly basic, because in water solution washing soda undergoes partial (reversible) exchange with water according to the equation:

$Na_2CO_3 + H_2O \rightleftarrows NaHCO_3 + NaOH$

(The symbol "⇄" means the reaction is reversible, so there always is just a small amount of NaOH in the washing soda solution, so it is not dangerous to handle).

Now you know how to obtain both an acid and a base for your experiment. Happy experimenting!

P.S. When we will be grading this problem, we will pay a special attention to the description of your experiment and the results you obtained. The more convincing your results will be (i.e. the more reliable and more visible the color change is) the more points you get. And, please, don't forget about control: it is highly desirable that you

provide the pictures for the same solution before an acid/base has been added and after that.

Hint:

No hints for this month Chemistry

Solution:

Indicators are, as a rule, organic dyes that are either weak acids or weak bases. When a hydrogen ion (a proton) dissociates from them (or associates) that may affect the electronic structure of the dye molecule. If the part of the molecule that is affected due to proton association/dissociation is involved in the interaction of the molecule with visible light, that cause color change.

There are two major classes of natural compounds that are colored and interact with hydrogen ions (i.e. are slightly acidic or basic). First class is polyphenols (found it tea), and the second class are flavonoids (found in many fruits and vegetables, although not all vegetables are colored due to them; for example, the color of carrot is due to totally different dye).

That ,means, tea, some kinds of fruit juice, red wine, and similar foods can serve as a “pH-paper”. In particular, black tea turns dark-brown in basic media, and it becomes light-yellow in the presence of acids. Red cabbage juice, beet juice, red grape juice, and some other juices contain flavonoids, and they change color both in acids and base.

10 points:

A cargo spacecraft *Giordano Bruno* loaded with various supplies for the new Martian colony was hit by a meteorite on its way to Mars. The damage was relatively minor, but the oxygen tank was completely destroyed, that meant Ridley Scott, the spacecraft’s pilot and the only crew member had just few hour to find an new source of oxygen to breathe until the rescue ship arrived (he sent the SOS signal immediately after the accident, but it takes about a week for the rescue spaceship to arrive to *Giordano Bruno*). Upon having inspected his spacecraft’s inventory, Ridley found the auxiliary engine’s oxidizer tank to contain 38 kg of hydrogen peroxide (the main *Bruno’s* engine was powered by a nuclear reactor, so it contained no oxidizer).

In addition to that Ridley found a box with alkaline batteries in the cargo compartment of his ship, as well as several tons of glass wool, which colonists used as a thermal insulator.

Question #1. How can Ridley produce oxygen from these materials (explain the most technically feasible way to do that)?

Question #2. Will these materials be sufficient to produce enough oxygen for Ridley before the rescue mission’s arrival?

To answer the second question, assume the following:

- the only reaction that produces energy in Ridley's body is glucose oxidation (that is a reasonable assumption):

$C_6H_{12}O_6 + 6O_2 \rightarrow 6CO_2 + 6H_2O$

- the heat of combustion of glucose is -2800 kJ/mol (that means combustion of 180 grams of glucose according to the above equation produces 2800 kJ of energy;180 grams of glucose is 1 mole of glucose; a mole is the amount of any substance measured *in grams* that is numerically equal to its molecular weight in Daltons)
- The minimal amount of energy our body has to produce to support metabolism is 100 kJ per kilogram per day.
- Ridley's weight is 70 kilograms.

Hint:

No hints for this month Chemistry

Solution:

Hydrogen peroxide can be a good source of oxygen, because it decomposes in the presence of salts or oxides of some transition metals. Alkaline batteries contain manganese dioxide, and it is a good catalyst of hydrogen peroxide decomposition.

$2\ H_2O_2 \rightarrow H_2O + O_2$

That means Ridley can produce some amount of oxygen. How much of oxygen can he produce? There are 38 kg of hydrogen peroxide, and the hydrogen peroxide used as a fuel has about 90% or more (it is being used rarely because it is not too stable, but let's assume future engineers learned how to prevent its destabilisation). Molar mass of hydrogen peroxide is 32*2 + 2 = 34 g. That means we have about 38000/34 = 1118 moles of peroxide (assuming it is 100%). Two molecules of H_2O_2 are needed to make one molecule of oxygen, which means maximal amount of oxygen Ridley can make is 559 moles.

Now we have to estimate the amount of oxygen Ridley needs. His organism consumes glucose, but it is not a combustion, because it occurs according to a totally different mechanism. However, the good thing about the law of conservation of energy is that, independent on the the mechanism of the reaction, the thermal effect of two reactions is the same if the starting materials and the products are the same. That means, combustion of glucose in oxygen, and transformation of glucose in Ridley's body

produce the same amount of energy, so we can use the thermal effect of glucose combustion to estimate the amount of oxygen Ridley needs.

Ridley needs 100 kJ * 70 = 7000 kJ of energy daily, or 49000 kJ weekly. To produce 2800 kJ, six moles of oxygen are needed. That means, to produce 49000 kJ, 6*49000/2800, 105 moles of oxygen are needed, which is much less than Ridley can produce from hydrogen peroxide he has.

BIOLOGY

5 points: There is a raging debate as to whether dogs or cats are more intelligent. Certainly dogs and cats have different types of behavior, but which is more "intelligent" depends upon how we define the term "intelligence." Imagine that you came across an extraterrestrial life form. What specific procedures and criteria would you use to determine which of the life form's behaviors were "hard-wired" versus which behaviors displayed "intelligence?"

Answer: Minimum criteria for intelligence should be defined as behavior adaptive to environmental changes.

10 points: DNA consists of two strands (*chromosomes*) lying side-by-side and winding around each other like a spiral staircase. The rungs that connect them are made of two buddies. Each buddy consists of one of four different chemicals (*nucleotides*), nicknamed A, T, G, and C. A gene is nothing but a little segment of the DNA thread containing several thousand nucleotides on one chromosome. The sequence in which the nucleotides are lined up on a gene tells the cell how to make a particular protein. The buddy system means that G is always with C and A is always with T. So if the DNA staircase gets unzipped in the middle and if we had only half of the staircase, we could tell exactly how the other half needs to look, and use new nucleotides (A, T, C, G) to fill in the other half. This property allows our genes not only to divide into two other genes, but also allows them to copy.

Exactly on the spot where a gene begins, the DNA staircase can be opened up a bit. The halves of the rung then become accessible. Now special proteins, called *polymerases*, start doing their job. They always attach a T to an A, an A to a T, a C to a G, and a G to a C. This way a fresh thread of tightly connected nucleotides appears—something called *messenger RNA*. RNA does not contain the nucleotide nicknamed T. Instead, it contains one called U, which takes the place of T. The messenger contains a true copy of the gene—the plan for making a particular protein. The messenger RNA thread leaves the nucleus of the cell with its message. It then goes to the protein factories in the cell, the *ribosomes*, where the messenger RNA's instructions can be carried out. *Now we can understand why different cells can make different proteins (e.g., hair, skin, eyes, muscles, blood) even though the same genes are present in everywhere.* It all depends upon whether or not a messenger RNA copy of a particular gene is made. To switch ON a gene means making a copy of it. For

instances, red blood cells would switch ON the gene for hemoglobin by making a messenger RNA copy of it. Skin cells would switch OFF the hemoglobin gene by simply not making a copy of it.

While turning genes on and off is necessary for healthy functioning in a body, the process can also go wrong. A common type of epigenomic modification is called *methylation*. Methylation involves attaching small molecules called methyl groups, each consisting of one carbon atom and three hydrogen atoms, to segments of DNA. When methyl groups are added to a particular gene, that gene is turned off or silenced, and no protein is produced from that gene. Because errors in the epigenetic process, such as modifying the wrong gene or failing to add a compound to a gene, can lead to abnormal gene activity or inactivity, they can cause genetic disorders. Conditions including cancers, metabolic disorders, and degenerative disorders have all been found to be related to epigenetic errors.

Pick one disease, and explain how errors in turning genes on or off leads to problems.

Answer:

https://www.nichd.nih.gov/health/topics/epigenetics/conditioninfo/Pages/impact.aspx

http://www.nature.com/scitable/topicpage/epigenetic-influences-and-disease-895

http://www.nature.com/scitable/content/epigenetic-diseases-and-their-causes-and-symptoms-37397

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:

In computer science, a set is a collection that stores unique items. We already saw 2 collections: array and list. Why do we need another one? Well, sets have some interesting properties:

1. They guarantee that the items you store in them will be unique, i.e. one of each. Here's an example:
array: [a, b, c]
list: a → b → c
set: {a, b, c}
If you add 'c' to them, you'll get this:
well, to be precise, let's add to the end of array and list. For sets it's just "add" (we'll discuss later),
[a, b, c, c]
a → b → c → c
{a, b, c} – because we cannot have two 'c'.

2. If you want to see if, for example, 'b' is in the collection then for arrays and lists you don't have any choice other than scan the (entire, in worst case) group to see if it's there. For sets it's just a single operation, i.e. it's fast.

3. Insertion and deletion is fast.

For example, if you want to add 'd' in the middle of array then you have to:
a. allocate new memory of size n+1, where n is the old size;
b. copy all elements up to where you want to insert d into the new array;
c. copy d;
d. copy all the rest of elements from old to new array;
e. free the memory taken by old array.

For list you have to:
a. start from the beginning of list and traverse the list until you find the spot you want to insert d. This may take some time if the list is large.
b. reassign the "next" link from the previous element to this one ('d');
c. set the "next" link of 'd' element to the next item.

For sets you just insert ‘d’. It’s kind of 1 operation.

Deletion is similar.

However, sets have a drawback, too. They don’t have order. For example, if you start adding letters to a set: {} → {a} → {a, b} → {a, b, c} and then print it, you may see: b, a, c.

5 points:

Implement a proof-of-concept spell-checker. Your program should work as follows:

1) Get user input to populate the dictionary
    a) For example, ask user how many words their dictionary has, and then let the user enter those 10 words.
2) Get string from user that needs to be spell-checked.
3) Break string into individual words
4) Check if each word is in the set, and if it is not, let the user know that that word is “misspelled”

Solution:

Java:

```
/*
  Implement a proof-of-concept spell-checker. Your program should work as follows:
  1) Get user input to populate the dictionary
    a) For example, ask user how many words their dictionary has, and then let the user enter
those 10 words.
  2) Get string from user that needs to be spell-checked.
  3) Break string into individual words.
  4) Check if each word is in the set, and if it is not, let the user know that that word is
"misspelled".
*/

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashSet;
import java.util.Set;

public class SpellCheckerJ {
  private int dictSize;
  private Set<String> dictionary = new HashSet<String>();
  private String text;

  public void inputDictionary() throws IOException {
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    System.out.print("How many words does your dictionary have? ");
    String line = br.readLine().trim();
    dictSize = Integer.parseInt(line);
```

```
    for(int i=1; i<=dictSize; i++) {
      System.out.print("Enter a dictionary word # "+i+": ");
      String word = br.readLine().trim();
      dictionary.add(word.toLowerCase()); // let's make it case insensitive
    }
  }

  public void inputText() throws IOException {
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter some text to spell check.");
    text = br.readLine().trim();
  }

  public boolean check() {
    boolean ret = true;

    String[] words = text.split("\\W+"); // by a non-word character, e.g. space, comma, etc.
    for(String word : words) {
      if(!dictionary.contains(word.toLowerCase())) { // let's make it case insensitive
        System.out.println("The word '"+word+"' is not found in the dictionary.");
        ret = false;
      }
    }

    return ret;
  }

  public static void main(String[] args) throws IOException {
    SpellCheckerJ checker = new SpellCheckerJ();
    checker.inputDictionary();
    checker.inputText();
    checker.check();
    System.out.println("end.");
  }
}
```

Python:

```
"""
  Implement a proof-of-concept spell-checker. Your program should work as follows:
  1) Get user input to populate the dictionary
    a) For example, ask user how many words their dictionary has, and then let the user enter
those 10 words.
  2) Get string from user that needs to be spell-checked.
  3) Break string into individual words.
  4) Check if each word is in the set, and if it is not, let the user know that that word is
"misspelled".
"""

import sys
import re

class SpellChecker:
  dictSize = 0
  dictionary = set()
  text = ""
```

```
  def inputDictionary(self):
    print("How many words does your dictionary have? ")
    self.dictSize = int(sys.stdin.readline().strip())

    for i in range(self.dictSize):
      print("Enter a dictionary word # %d: " % (i+1))
      word = sys.stdin.readline().strip()
      self.dictionary.add(word.lower()) # let's make it case insensitive

  def inputText(self):
    print("Enter some text to spell check.")
    self.text = sys.stdin.readline().strip()

  def check(self):
    ret = True

    words = re.split("\\W+", self.text); # by a non-word character, e.g. space, comma, etc.
    for word in words:
      if not word: # python adds empty strings in split(), so ignore
        continue
      if word.lower() not in self.dictionary: # let's make it case insensitive
        print("The word '%s' is not found in the dictionary." % word)
        ret = False

    return ret

if __name__ == "__main__":
  checker = SpellChecker()
  checker.inputDictionary();
  checker.inputText();
  checker.check();
  print("end.")
```

10 points:

You may have noticed that you didn’t need to use sets to implement the spell-checker above. For instance, you could have used arrays or lists to store the dictionary. In this problem, you will investigate whether sets offer speed improvements over the more elementary method.

Implementing a Microbenchmark test

In this problem we will motivate the use of sets over just arrays or lists. You will create a very large list and a very large set, and you will time how long it takes for the computer to process the same operation with these two methods. Here is the outline of the program you want to write. It really isn't difficult.

Set up:

1) Make an array or list of 10,000 random 10-digit numbers (search online if you're not sure how to do this)
2) Feed this array into a set. [by looping over each list element and running .add()]

The test to determine how fast lists are:
1) Start a timer. (search google for "how to time things in [python/java..]" if you don't know how to implement a timer)
2) Repeat the following loop N times (make N large enough for the test to take a few seconds):
- Create a random 10-digit number
- Check if the number is in the list by looping over the list manually or using a build-in method for lists (like '.contains()')
3) End timer.

Repeat the same thing for sets. Which one took longer? Is using sets worth it?

Solution:

Java:

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.LinkedList;
import java.util.List;
import java.util.Random;
import java.util.Set;

public class BenchmarkJ {
  public static void main(String[] args) {
    int n = 10_000; // collection size
    int m = 100_000; // number of tests
    Random r = new Random();
    int min = 1_000_000_000; // minimum 10 digit number
    int max = Integer.MAX_VALUE; // == 2,147,483,647

    Set<Integer> set    = new HashSet<Integer>();
    List<Integer> list  = new LinkedList<Integer>();
    List<Integer> array = new ArrayList<Integer>(); // resizable-array implementation of the
List interface

    // initialize
    for(int i=0; i<n; i++) {
      int k = r.nextInt((max - min) + 1) + min;
      set.add(k);
      list.add(k);
      array.add(k);
    }

    // test
```

```
    long start = System.currentTimeMillis();
    for(int i=0; i<m; i++) {
      int k = r.nextInt((max - min) + 1) + min;
      boolean b = set.contains(k);
    }
    long end = System.currentTimeMillis();
    System.out.printf("Set took %d ms\n", end-start);

    start = System.currentTimeMillis();
    for(int i=0; i<m; i++) {
      int k = r.nextInt((max - min) + 1) + min;
      boolean b = list.contains(k);
    }
    end = System.currentTimeMillis();
    System.out.printf("List took %d ms\n", end-start);

    start = System.currentTimeMillis();
    for(int i=0; i<m; i++) {
      int k = r.nextInt((max - min) + 1) + min;
      boolean b = array.contains(k);
    }
    end = System.currentTimeMillis();
    System.out.printf("Array took %d ms\n", end-start);
  }
}
```

Python:

```
import numpy as np
import random
import time

n = 10000 # collection size
m = 100000 # number of tests
mn = 1000000000 # minimum 10 digit number
mx = 2147483647

st   = set()
lst  = []
arr  = np.ndarray(n)

# initialize
for i in range(n):
  k = random.randint(mn, mx)
  st.add(k)
  lst.append(k)
  arr[i] = k

# test
start = time.clock()
for i in range(m):
  k = random.randint(mn, mx)
  if k in st:
    b = 1
end = time.clock()
print("Set took %f s" % (end-start))
```

```
start = time.clock()
for i in range(m):
  k = random.randint(mn, mx)
  if k in lst:
    b = 1
end = time.clock()
print("List took %f s" % (end-start))

start = time.clock()
for i in range(m):
  k = random.randint(mn, mx)
  if k in arr:
    b = 1
end = time.clock()
print("Array took %f s" % (end-start))
```