Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Übungen zur Vorlesung Informatik II (D-BAUG) FS 2017

D. Sidler, F. Friedrich

http://lec.inf.ethz.ch/baug/informatik2/2017

Solution to exercise sheet # 2 **27.2.2017 – 7.3.2017**

Please provide your solutions to the tasks using the ETH Codeboard submission system.

For the exercises please use the definitions from the lecture for $\mathcal{O}$, Ω and Θ. We have defined

$$\mathcal{O}(f) = \{g : \mathbb{N} \to \mathbb{R} \mid \exists c > 0, n_0 \in \mathbb{N} : 0 \leq g(n) \leq c \cdot f(n)\ \forall n \geq n_0\},$$
$$\Omega(f) = \{g : \mathbb{N} \to \mathbb{R} \mid \exists c > 0, n_0 \in \mathbb{N} : 0 \leq c \cdot f(n) \leq g(n)\ \forall n \geq n_0\}, \text{and}$$
$$\Theta(f) = \Omega(f) \cap \mathcal{O}(f).$$

Problem 2.1. Big O notation

Complete the following table. For each function $f(n)$ determine $\mathcal{O}$, the upper bound of the growth rate.

$f(n)$	$f \in \mathcal{O}(?)$
$3n^2 + 5$	
$7n$	
$3n + 2$	
$log_2(n) + 5$	
$n * n$	
$(n * n + 1) * n * n/2$	

Submission link: https://codeboard.ethz.ch/inf2baugex02t01

Solution of Problem 2.1.

$f(n)$	$f \in \mathcal{O}(?)$
$3n^2 + 5$	$\mathcal{O}(n^2)$
$7n$	$\mathcal{O}(n)$
$3n + 2$	$\mathcal{O}(n)$
$log_2(n) + 5$	$\mathcal{O}(\log n)$
$n * n$	$\mathcal{O}(n^2)$
$(n * n + 1) * n * n/2$	$\mathcal{O}(n^4)$

Problem 2.2. Asymptotic Growth.

Sort the following functions from left to right such that: if function f is left to function g, then $f \in \mathcal{O}(g)$. Example: n^3, n^7, n^9 are in a correct order ($n^3 \in \mathcal{O}(n^7)$, $n^7 \in \mathcal{O}(n^9)$).

$$n^5 + n,\ \log(n^4),\ \sqrt{n},\ \binom{n}{3},\ 2^{16},\ n^n,\ n!,\ \frac{2^n}{n^2},\ \log^8(n),\ n \log n.$$

Submission link: https://codeboard.ethz.ch/inf2baugex02t02

Solution of Problem 2.2.

Note that $\binom{n}{3} = \frac{1}{6}(n-2)(n-1)n$. The only correct order is

$$2^{16},\ \log(n^4),\ \log^8(n),\ \sqrt{n},\ n \log n,\ \binom{n}{3},\ n^5 + n,\ \frac{2^n}{n^2},\ n!,\ n^n.$$

Problem 2.3. The set $\Theta(g)$.

Give a counterexample that demonstrates that the right-hand side of the following equation does *not* hold.

$$\Theta(f) = \{g : \mathbb{N} \to \mathbb{R}^+ \mid \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N}, \forall n \geq n_0 : g(n) = c \cdot f(n)\}.$$

Give a correct definition of the set $\Theta(f)$ as compact as possible (i.e. with the fewest possible parameters and quantifiers) analogously to the definitions above for sets $\mathcal{O}(f)$ and $\Omega(f)$.

Submission link: https://codeboard.ethz.ch/inf2baugex02t03

Solution of Problem 2.3. The function $g(n) = n+1$ is clearly in $\mathcal{O}(f)$ and $\Omega(f)$ for $f(n) = n$, but doesn't satisfy the equation $g(n) = c \cdot f(n)$.

A correct compact definition is:

$$\Theta(f) = \{g : \mathbb{N} \to \mathbb{R} \mid \exists a > 0,\ b > 0,\ n_0 \in \mathbb{N} : 0 \leq a \cdot f(n) \leq g(n) \leq b \cdot f(n)\ \forall n \geq n_0\}.$$

But this can be done even more compact. If the definition holds for an a and b, we can find a c such that $b \leq c$ and $\frac{1}{c} \leq a$. Using this c we can give a more compact definition

$$\Theta(f) = \{g : \mathbb{N} \to \mathbb{R} \mid \exists c > 0,\ n_0 \in \mathbb{N} : 0 \leq \frac{1}{c} \cdot f(n) \leq g(n) \leq c \cdot f(n)\ \forall n \geq n_0\}.$$

Problem 2.4. Programming Exercise – Fair Dice

Open the task description here: https://codeboard.ethz.ch/inf2baugex02t04. In this task you are implementing a fair dice. In the main function a random generator is instantiated:

```
Random generator = new Random(0);
```

You can use this random generator to get random numbers in the interval $[0, 1)$ by calling the function `generator.nextDouble()`.

After you have obtained a random number you have to map it to the numbers 1-6 of the dice. Use the following mapping:
$[0, \frac{1}{6}) \to 1$
$[\frac{1}{6}, \frac{2}{6}) \to 2$
$[\frac{2}{6}, \frac{3}{6}) \to 3$
$[\frac{3}{6}, \frac{4}{6}) \to 4$
$[\frac{4}{6}, \frac{5}{6}) \to 5$
$[\frac{5}{6}, 1) \to 6$

As you can see that `main()` function takes an integer `x` as input and then executes the for loop `x` times. This means that it "throws" the dice `x` times and `x` numbers are printed out.

You can test and submit your program by un-commenting the annotation `@RunTests`.

Solution of Problem 2.4.

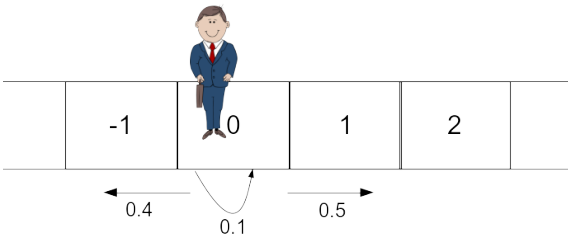
```
/**
 * Main class of the Java program.
 *
 * For TESTING and SUBMITTING: Uncomment the @RunTests annotation
 * (Remove the two slashes at the beginning of line ~11)
 *
 */
import java.util.Scanner;
import java.util.Random;

@RunTests
public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner (System.in);
        int x = input.nextInt();
        //Get Random generator with seed 0
```

```
        Random generator = new Random(0);
        // Get x random integers between 1-6
        for (int i = 0; i < x; i++) {
           // Get random number between [0,1) by calling generator.nextDouble()
           double sample = generator.nextDouble();
           // Map random to integers 1,2,3,4,5,6
           int rand = (int) (sample*6) + 1;
           // Print out integer number
           System.out.println(rand);
        }
    }
}
```

Problem 2.5. Programming Exercise – 1D Ranom Walk

Open the task description here: https://codeboard.ethz.ch/inf2baugex02t05. In this task you implement a random walker. The walker walks to the left (field -1) with probability 0.4, stays on the current field with probability 0.1 and walks to the right (field $+1$) with probability 0.5. The figure below illustrates this.



As you can see in the main function in Main.java the random walker starts on field 0. Complete the function by implementing the following steps.

1. Call `generator.nextDouble()` to get a random number between $[0, 1)$.
2. Use this random number to adapt the field value like this:
   $[0, 0.4) \rightarrow field = field - 1$
   $[0.4, 0.5) \rightarrow field = field$
   $[0.5, 1) \rightarrow field = field + 1$
3. After calculating the new field value print it out. For example if the field value is "2" then the print out should be "Feld: 2".

Now test your program by un-commenting the annotation `@RunTests`. Once you pass the test you can submit your program.

Solution of Problem 2.5.

```
/**
 * Main class of the Java program.
 *
 * For TESTING and SUBMITTING: Uncomment the @RunTests annotation
 * (Remove the two slashes at the beginning of line ~11)
 *
 */
import java.util.Scanner;
import java.util.Random;
```

```
@RunTests
public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner (System.in);
      int x = input.nextInt();
      //Get Random generator with seed 0
      Random generator = new Random(0);
      //Start on field 0
      int field = 0;
      // The walker walks x times
      for (int i = 0; i < x; i++) {
         // Get number between [0,1) by calling generator.nextDouble()
         double sample = generator.nextDouble();
         // Modify field according to mapping in the exercise
         if (sample < 0.4) {
            field--;
         }
         else if(sample >= 0.5) {
            field++;
         }
         // print out field value
         System.out.println("Feld: "+field);
      }
   }
}
```