Adrian Willi

Zurich University of Applied Sciences

School of Engineering

IDP Institute of Data Analysis and Process Design

MSE | MASTER OF SCIENCE IN ENGINEERING

Optimization HS22

Dr. Peter Fusek (fuse@zhaw.ch)

Exercises Series 1

Issue date: 19th/21st September 2022

Exercise 1

Read the Chapter 1 – *Introduction* and Chapter 2 – *Overview of the Operations Research Modeling Approach* of the book *Hillier, F.S., Lieberman, G.J., Introduction to Operations Research, 7th Edition, McGraw-Hill, 2001.*

Exercise 2

The construction of a lecture timetable can be mathematically formulated (simplified) as an assignment of a lecturer, a time slot and a room to every lecture.

Find an upper bound for the number of possible timetables if there are 500 lectures per week, 100 lecturers, 40 time slots per week and 15 rooms.
(An upper bound for a value we are looking for is a number which cannot be exceeded by that value. Typically, one is looking for good, i.e. low upper bounds.)

To simplify the task we assume that the time-tables are identical for different weeks and, basically every lecturer, every time slot and every room can be assigned to any lecture.

How many years would it take a modern computer to enumerate all possible time-tables if we assume that the computer can enumerate 1 million time-tables per second?

Exercise 3

A naïve method for sorting a list of n numbers can be formulated as follows: in Step 1, determine the maximum of all n numbers and erase it from the list; in Step 2 determine the maximum of the remaining $n - 1$ numbers etc. (Notice that there exist much more efficient sorting methods.)

What is the size of the largest list which can be sorted in one minute by a modern computer using the above method?

Here we assume that the computer needs exactly n "elementary operations" to find (and erase) the maximum in a list consisting of n numbers. An elementary operation in this context essentially consists of comparing two numbers and saving the bigger one of them. Further, we assume that the computer can perform 1 million of such "elementary operations" per second.

Exercise 2

First we aim for one lecture only. In this case we would have 1 lecturers, s slots and r rooms. Thanks to the simplified problem we can write it down in the following way $1 \cdot s \cdot r$. Now we just expand it to all the lectures and we obtain the upper bound

$$= (1 \cdot s \cdot r)^{500}$$
$$= (100 \cdot 40 \cdot 15)^{500}$$
$$\approx 1.1902 \cdot 10^{2'385}$$

Knowing that our computer can enumerate 1 million tables per second we get

$$\frac{1.1902 \cdot 10^{2'385} \text{ timetables}}{10^6 \text{ timetables/second}} = 1.1902 \cdot 10^{2'383} \text{ seconds} \approx 3.774 \cdot 10^{2'375} \text{ years}$$

Exercise 3

For the first maximum we need n operations. For the next $n-1$, and so on. Therfore, the number of operations needed in total is the sum of them.

$$\sum_{i=0}^{n} n-i = n + (n-1) + (n-2) + \cdots + 1 = \frac{n(n+1)}{2}$$

Our modern computer is able to perform $60 \cdot 10^6 = 6 \cdot 10^7$ operations per minute. So we need to solve the following equation

$$\frac{n(n+1)}{2} = 6 \cdot 10^7$$
$$n(n+1) = 12 \cdot 10^7$$
$$n^2 + n = 12 \cdot 10^7$$
$$n^2 + n - 12 \cdot 10^7 = 0$$

Solving for n gives us $n_1 = 10'953$ and $n_2 = -10'954$. We discard n_2 thus it's negative. So our modern computer is able to process a list with 10'953 elements.