Easy and Hard Problems in Computer Science

By Dr. Shahid Hussain

Reviewed by Mohammad Ibrahim Ali

2 September 2019

This report reviews the first guest lecture of the CS Freshmen Seminar, delivered by Dr Shahid Hussain, *Program Director* and *Assistant Professor Computer Science* at *Habib University*. The lecture aimed to highlight how a computational problem is classified.

The lecturer stated that computational problems are differentiated by their level of difficulty. He gave the example of **Travelling Salesperson Problem (TSP)**, in which six points were pointed on a map. The problem was to find the shortest route to travel through these points while starting and ending at the same point. Since there was no other way except to check each route and calculating the distance or time, it concluded that the increase in the number of points could turn the problem harder. Hence, it was classified as a **hard problem**. Contrary to this, the idea of an **easy problem** was discussed, which was defined as a problem that doesn't require more time to solve with the increase in data, as in sorting numbers.

The lecturer also highlighted some interesting facts like,

- Even a fast computer that could compute 1 million operations per second would require 2 million years to solve a TSP problem with 25 points.
- Hard problems could be not-very hard, and easy problems could be not-very easy, which leads to another classification.

The lecture concluded to the idea that a programmer should be able to detect the problem as easy or hard and should prefer easy problems, to bring productivity in his work.

References

`http://bit.ly/2lOsUYI`