

Computationally Hard Problemes

Course 02249

English – 7.5 Points – Fall 13-weeks

Lectures and assignments some of which are mandatory

Written examination

Paul Fischer

Title: Computationally Hard Problemes

Aim: To enable the students to identify a problem as being hard. To introduce the students to the powerful concept of randomization.

Content: There are many important problems in Computer Sciences and other areas for which optimal solutions can be found but the time to do so is so long that it is not practical. Examples of such problems are optimal assignment of tasks to processors, finding an optimal tour for the local postman, optimal packing of a suitcase, optimizing the payload for a space mission or selecting a profitable stock portfolio.

In the course we will first specify what we mean by a “computationally hard” problem. Then some examples of such that problems are presented and methods are described to identify a problem as being hard. In order to become accessible to an analysis real world problems have to be converted into abstract mathematical models which capture the essence of the original problem.

We shall then look at problems where randomization helps in solving them. This means that there are fast algorithms for solving the problem that use “coin flips”, i.e. a random number generator. Hence repeatedly running such an algorithm on the same input might lead to different solutions. We shall consider problems where a “good” solution will be found with high probability by randomized algorithms.