Algorithms – handout 1

Uriel Feige *

October 21, 1999

A variety of algorithms in different areas will be discussed, including graph algorithms, algebraic algorithms, combinatorial optimization, string processing, and number theory. Various algorithmic techniques will be demonstrated, such as greedy algorithms, dynamic programming, divide and conquer, duality theory and randomization. This is a basic graduate course – it is anticipated that this course will be followed by a more advanced one in the Spring Semester.

The students are expected to be familiar with algorithms, data structures, probability theory and linear algebra, at an undergraduate level.

Homework assignments are very important to the process of learning, and also effect the final grade to a large extent. Try to do the best you can. They should be handed in within two weeks.

There are many good books on algorithms. The book "Introduction to Algorithms" by Cormen, Leiserson and Rivest can serve as a useful textbook.

Today we shall discuss Quicksort. Note that we will give a simpler analysis of its running time than that given in CLR.

Homework: In the randomized version of quicksort, the splitting element is chosen at random. A modified method chooses three elements at random and uses their median as the splitting element. This involves three comparisons in the process of choosing the splitting element. Two of them classify two other elements, and hence would have been done anyway. The purpose of this homework assignment is to show that the expected number of comparisons in the modified quicksort algorithm is smaller than that of the original randomized one (without actually computing what the new expectation is).

1. For $\alpha < 1$, approximate the probability that the modified procedure produces a splitting element that falls within the middle αn elements of the input. Compare this with the probability that randomized quicksort produces such an alement. Use this to argue (intuitively) that the modified procedure is expected to use less comparisons.

2. Show formally that the expected number of comparisons in the modified algorithm is smaller than the expected number of comparisons in the original algorithm. You may ignore here the extra comparison introduced in the process of choosing the median. (Any formal proof would do. However, here is a hint for the proof that I have in mind: analyse the modified quicksort using the framework used in class for randomized quicksort. At some point in the analysis, use concavity of the function $x(1-x)$.)

*Zyskind 247. phone 3364. `feige@wisdom.weizmann.ac.il`