COMP/MATH 3804 – Design and Analysis of Algorithms I

Fall 2015

Assignment 5

due on November 25, 2015

Problem 1. Modify the Mergesort algorithm as follows: instead of splitting the input array into two roughly equal parts, split it into three roughly equal parts. Recursively call the modified Mergesort algorithm on each part, and then merge the three resulting sorted sub-arrays into one final sorted array. What is the running time of this algorithm? (You may assume the size of the array to be a power of 3.)

Problem 2. The *continuous knapsack problem* is similar to the knapsack problem without repetition, except that now we are allowed to "break" items and put fractional parts of them in the knapsack. Describe (in English) an efficient greedy algorithm that finds an optimal solution to a given instance of the continuous knapsack problem. *All pseudo-code will be ignored.*

Problem 3. Given an undirected graph $G = (V, E)$ and an integer k, describe (in English) an algorithm that finds the largest (not necessarily connected) subgraph H of G such that each vertex in H has degree at least k, or determines that such a subgraph does not exist. The algorithm should run in time $O(|V| + |E|)$. *All pseudo-code will be ignored.*

Problem 4. You are tasked to sort an array of n distinct integers, under the assumption that $n - \lfloor\sqrt{n}\rfloor$ of its elements are already in their final positions. Describe (in English) a linear-time algorithm to sort such an array. *All pseudo-code will be ignored.*