Algorithms Qualifying Exam Sample Exam 1

Fall 2014

In the KNAPSACK problem, one is given a set A of n items, a knapsack of capacity W, and an integer k. Item i has a weight w_i and a value v_i. The problem is to determine if there is a subset $B \subseteq A$ of items that fits into the knapsack (i.e., $\sum_{i \in B} w_i \leq W$) and has value at least k (i.e., $\sum_{i \in B} v_i \geq k$). The problem is well known to be NP-Complete

In this problem, we will consider the *fractional* knapsack problem. In this variant of the problem, one can select a fraction of an item (rather than having to completely include the item or completely leave it out). The weight and value accumulated from the fractional item is scaled accordingly. For example, if one selects $\frac{1}{2}$ of item i, then weight $\frac{w_i}{2}$ is added to the knapsack and we gain a value of $\frac{v_i}{2}$. The problem is to determine if it is possible to accumulate value of at least k subject to the constraint that the weights sum to at most W.

Propose algorithms for the fractional knapsack problem that solve the problem (exactly or approximately) using one or more approaches (such as divide-and-conquer, greedy, dynamic programming, graph traversal, shortest path, max flow, etc). The goal should be to clearly demonstrate strong critical thinking skills regarding algorithms. Argue why a some approach may be reasonable, why some approaches may not be reasonable, and provide a critique of your solution.