MATH182 DISCUSSION 5 WORKSHEET

1. Knapsacks are pretty small; let's get a wagon instead

The *knapsack problem* is a well-known combinatorial optimization problem: Given a set of items with specified weights and values, how do you choose items so as to maximize the total value while keeping the total weight below a given bound? More formally, we are given a list of (positive integer) weights $w_1, \ldots, w_n$, a list of (positive) values $v_1, \ldots, v_n$, a set of permitted quantities Q, and a weight limit W, and we wish to find quantities $q_1, \ldots, q_n \in Q$ which maximize $\sum_{i=1}^{n} q_i v_i$ subject to the constraint $\sum_{i=1}^{n} q_i w_i \leq W$.

There are several common choices for the permitted quantities Q. We will investigate the cases $Q = \mathbb{N}$ (the *unbounded knapsack problem*), $Q = \{0, 1\}$ (the *0-1 knapsack problem*), and $Q = [0, 1]$ (the *fractional* or *continuous knapsack problem*). All of them have the optimal substructure property:

Exercise 1. Prove that if $q_1, \ldots, q_n$ is an optimal solution to the unbounded knapsack problem for a given weight limit W, and $q'_1, \ldots, q'_n \in Q$ are such that $q'_i \leq q_i$ for $i = 1, \ldots, n$, then $q'_1, \ldots, q'_n$ is an optimal solution for the weight limit $W' = W - \sum_{i=1}^{n}(q_i - q'_i)w_i$. This shows that UKP has optimal substructure.

Exercise 2. Prove that if $q_1, \ldots, q_n$ is an optimal solution to the fractional or 0-1 knapsack problem for a given weight limit W, and $j \leq n$, then $q_1, \ldots, q_j$ is an optimal solution optimal solution for the weights $w_1, \ldots, w_j$ and and values $v_1, \ldots, v_j$, and weight limit $W' = W - \sum_{i=j+1}^{n} q_i w_i$. This shows that these knapsack problems have optimal substructure.

Exercise 3. Give examples to show that the 0-1 and fractional knapsack problems do not have the optimal substructure property we proved for the unbounded knapsack problem.

One intuitively-reasonable strategy for the knapsack problem is to choose the items with the most value per unit weight first. This works for the fractional knapsack problem:

Exercise 4. Show that the fractional knapsack problem has the greedy-choice property. That is, show that if $i \in \{1, \ldots, n\}$ is such that $\frac{v_i}{w_i}$ is maximal, and $q \in [0, 1]$ is such that $qw_i \leq W$, then there is an optimal solution to the fractional knapsack problem with $q_i \geq q$.

Exercise 5. Write a greedy algorithm for the fractional knapsack problem based on the above strategy. What is its running time?

The other problems are not solvable by such a greedy algorithm:

Exercise 6. Construct an instance of the 0-1 or unbounded knapsack problem for which the optimal solution does not include the item with the highest value to weight ratio.

We must therefore use a different, slower approach. We have already seen that the unbounded and 0-1 knapsack problems have optimal substructure, which we can translate into dynamic programming algorithms:

Exercise 7. Write a dynamic programming algorithm for the unbounded knapsack problem with running time $O(nW)$.

Exercise 8. Write a dynamic programming algorithm for the 0-1 knapsack problem with running time $O(nW)$.

2. No late homework accepted

You have n homework assignments to complete, with due dates $d_1, \ldots, d_n$. The ith homework is worth a_i points if completed by the due date, and zero otherwise. Each assignment will take you one day to complete, and you intend to do all of your homework, even if the deadline has passed. You wish to find an order in which to do your homework so as to maximize your total points.

Exercise 9. Show that this problem has optimal substructure with respect to choosing the first $k \leq n$ assignments. That is, if a given ordering is optimal for all n assignments, then the ordering of the first k is optimal for those k assignments.

Exercise 10. Write a greedy algorithm to solve this problem, and determine its running time.