| CS 218 | Homework 3, due 4/2/16 | Abhiram Ranade |
|---|---|---|

1. In class when we discussed the fractional knapsack problem, we glossed over an important point: we implicitly assumed that the value densities were distinct for all items: only then do we have a unique order of non-increasing densities. Suppose however, that some 2 items have same value density, i.e. for some $i, j$ we have $v_i/w_i = v_j/w_j$. Now, you will get a different solution if you pick $i$ first and $j$ later, as opposed to picking $j$ first and $i$ later. However, show that both these solutions must have the same value.

2. No one knows how to get an optimal solution to the knapsack problem in polynomial time. So people have tried to get solutions which give value *close* to the optimal value. In this problem you will see one approach that finds a solution of at least half the optimal value. (Admittedly, half is not terribly close to 1. However, later on we may see algorithms for getting closer).

   Suppose $C, W[1..n], V[1..n]$ is the knapsack instance. You can assume without loss of generality that all $W[i] \leq C$ (if not, throw out that item since it will not be used anyway). The algorithm for getting half the optimal value is:

   (a) Solve the fractional knapsack problem for $C, W[1..n], V[1..n]$.

   (b) We know that the solution will contain at most one item chosen fractionally, say item $j$, and a set of items $S$ chosen fully.

   (c) Consider two solutions to the (non-fractional, also called 0-1) knapsack problem: (A) consisting only of item $j$ chosen fully (B) consisting of set $S$.

   (d) Return the more valuable among the solutions (A),(B).

   Show that this algorithm indeed gives a solution of value at least 1/2 the value of the optimal solution to the 0-1 knapsack problem.

   The following fact is useful: The optimal solution to the fractional knapsack will have at least as much value as the optimal solution to the 0-1 knapsack. To prove this note that any solution to the 0-1 knapsack problem is also a solution to the fractional problem. Thus whatever value you can acheive in 0-1 knapsack can be acheived in the fractional problem; the fractional solutions may enable you to acheive something better.

3. I have $n$ jobs, respectively taking times $t_1, \ldots, t_n$. I must execute them in any order. I would like to minimize the average finishing time. Thus if job $i$ starts execution at time $t$, then its finishing time is $t + t_i$. In other words, the goal is to find a permutation $\pi$ and execute the jobs in the order $\pi(1), \pi(2), \ldots, \pi(n)$. Thus, the $i$th job executed in this order finishes at time $\sum_{j=1}^{i} t_{\pi(j)}$. So in other words you must minimize the quantity $\frac{1}{n} \sum_i \sum_{j=1}^{i} t_{\pi(j)}$

   Try to *derive* the greedy criterion for solving this problem. Specifically, assume that $\pi$ is an optimal permutation. Suppose you exchange the order of some pair of jobs which are being executed consecutively. Clearly, this should not decrease the average finishing time. Based on this see if you can get a condition on $\pi$.

   You may be able to prove more directly what the order must be; however, follow the directions above so that you understand the notion of perturbing the optimal solution and using that to derive the condition.