CMPSC 623 Problem Set 6.
by Prof. Honggang Zhang

Out: November 6, 2007
Due: November 13, 2007, before class.

Problem 1 Page 378, 16.1-2.

This proof is very similar to the proof for Theorem 16.1 in textbook.

Consider $S_{ij} = \{a_k \in S : f_i \leq s_k < f_k \leq s_j\}$.

Let a_m be the activity in S_{ij} with the latest start time, that is, $f_m = max\{s_k : a_k \in S_{ij}\}$. We first observe that S_{mj} is empty, as there is no other activity in S_{ij} that has a later start time than that of a_m.

We further observe that a_m is used in some max-size subset of mutually compatible activities in S_{ij}. In order to show this, first lets assume that we have A_{ij}, a max-size subset of mutually compatible activities of S_{ij} and lets order the activities in A_{ij} in monotonically increasing order of starting time. Let a_k be the last activity in A_{ij}. If $a_k = a_m$, we are done. Otherwise, construct a new subset $A'_{ij} = A_{ij} - \{a_k\} \cup \{a_m\}$. Clearly A'_{ij} is also a max-size solution for S_{ij} as all activities in A'_{ij} are mutually compatible and its size is the same as that of A_{ij}.

Based on these two observations, we see that to look for a max-size set solution for S_{ij}, we can simply choose the latest starting-time activity in S_{ij}, and then recursively solve a smaller subsubproblem S_{im}. This will give us a sequence of greedy choices when solving for $S_{0(n+1)}$. This sequence of greedy choices is an optimal solution for the original problem.

Problem 2 Page 384, 16.2-1.

Define a subproblem S_{ij} as the set of items from 1 to i and the knapsack has capacity j. Define A_{ij} as the optimal set of fractional items for subproblem S_{ij} and let $F_{ij} = \{k \in A_{ij} : 0 < f_k \leq 1\}$ be the set of corresponding optimal fractions. That is, a complete optimal solution is given by S_{ij} and A_{ij}. Let the corresponding optimal value be $knap(i, j)$. A_{ij} contains an optimal collection of items. An item k in A_{ij} has a certain fraction of its weight $f_k w_k$ used in computing the optimal value $knap(i, j)$. That is, $knap(i, j) = \sum_{k \in A_{ij}} v_k f_k$.

We would like to show this problem has the greedy-choice property, that is, a globally optimal solution can be arrived by making a locally optimal greedy choice. Specifically for this problem, we would like to show that the most valuable item in subproblem S_{ij} must be in some optimal set. Let item m be the most valuable item in S_{ij}, that is, $v_m = max\{v_k : 1 \leq k \leq i\}$. We sort all items in A_{ij} in terms of decreasing value, and pick the most valuable item k. We can always replace item k's weight $f_k w_k$ in the optimal knapsack with the same weight $f_m w_m$ (where $f_m = f_k w_k / w_m$) **if** $w_m \geq w_k$, or with weight w_m of item m plus weight $f_k w_k - w_m$ of item k **if** $w_m < w_k$. Then this new optimal solution for S_{ij} has a total value no less than that of optimal solution given by A_{ij} and F_{ij}. This shows the greedy-choice property, that is, we can always choose the current most valuable item and put it into the knapsack as much as we can, and then fill the rest of the knapsack by solving a smaller subproblem following a similar greedy stategy.

Problem 3 Page 384, 16.2-3.

Lets sort items in increasing weight $w_1 \leq w_2 \leq ... \leq w_n$, then as stated in the problem description, we have $v_1 \geq v_2 \geq ... \geq v_n$.

We can simply design a greedy algorithm in which we scan all items starting from item 1. Whenever we check item i, we put it into the knapsack if it can fit into the current remaining capacity of the knapsack, that is if $w_i \leq W - \sum_{k \in A_{i-1}} w_k$ where A_{i-1} is the optimal set of items when considering items from 1 to $i-1$.

This problem clearly has greedy choice property. Consider subproblem S_{ij} where we consider items from 1 to i and knapsack capacity is j. Let m be the most valuable item in S_{ij}. We can show that m must be in some optimal solution for S_{ij}. Let A_{ij} be one optimal set for subproblem S_{ij}. Let k be the most valuable item in A_{ij}. If $k = m$, then we are done. Otherwise, we can replace k with m to get $A'_{ij} = A_{ij} - \{k\} \cup \{m\}$. The total weight of items in A'_{ij} is no larger than that of A_{ij} (because $w_m \leq w_k$), but its value is no less than that of A_{ij}, so A'_{ij} must be an optimal solution. Thus, we have proved the greedy choice property.

Problem 4 Page 384, 16.2-4.

Suppose that there are m gas stations in total. Lets call the gas station closest to the starting point g_1, and label all other gas stations as $g_2, g_3, ...g_m$ where a larger index of a gas station indicates that the gas station is further away from the starting point. Let d_i denote the distance between g_i and g_{i-1}. d_1 denotes the distance between g_1 and the starting point. d_{m+1} denotes the distance between the destination and g_m. d_0 denotes the distance between the starting point and itself, so it equals 0.

We can design a simple greedy algorithm to solve this problem. The idea is like this:

```
i=0
while i < m
   d <- 0
   k <- i
   while d <= n AND k <= m   //stop if the total miles exceed n
       k <- k+1
       d <- d+d[k]
   end
   if d > n
       print "stop at gas station k-1"
       i <- k-1            //we start again at gas station k-1
       continue
   else if k > m
       exit while loop
end
```

This algorithm is greedy in the sense that we always drive as far as we can (if within n miles). This problem has greedy choice property. To see this, let g_p be the first furthest gas station we can reach using one full tank of gas. That is, p is the largest index i to maximize

$\{d_1 + d_2 + ... + d_i\}$ subject to constraint $d_1 + d_2 + ... + d_i \leq n$. We can show that p must be in one optimal solution. Let A_{1m} denote the optimal set of gas stations for the original problem and let k be the largest index gas station in in A_{1m}. If $k = p$, we are done. Otherwise, let $A'_{1m} = A_{1m} - \{k\} \cup \{p\}$. Clearly, g_k cannot be further away from starting point than g_p and we can drive from starting point to g_p without gas re-filling (based on the definition of g_p) , so A'_{1m} is also a valid solution. Also because $|A'_{1m}| = |A_{1m}|$, so we know A'_{1m} is also an optimal solution. After we first choose a locally best gas station p, then we can follow a similar argument to show we can always choose the current best gas station for the remaining sequence of subproblems. This completes our proof.