CSE 60111: Complexity and Algorithms
Homework 7

---

1. A cargo plane can carry a maximum weight of 100 tons and a maximum volume of 60 cubic meters. There are three materials to be transported, and the cargo company may choose to carry any amount of each, up to the maximum available limits given below.

   - Material 1 has density 2 tons/cubic meter, maximum available amount 40 cubic meters, and revenue $1,000 per cubic meter.
   - Material 2 has density 1 ton/cubic meter, maximum available amount 30 cubic meters, and revenue $1,200 per cubic meter.
   - Material 3 has density 3 tons/cubic meter, maximum available amount 20 cubic meters, and revenue $12,000 per cubic meter.

   (a) Write a linear program that optimizes revenue within the constraints. Note: You don't need to solve it, just set it up!

   (b) Now put the linear program into canonical form. Again, no need to solve.

   (c) Write the dual linear program. (In case I need to reiterate it: don't solve this dual LP, either.)

2. Consider this variant of the change-making problem, which we briefly discussed (in a different format) earlier: Given an unlimited supply of coins of denominations $x_1, x_2, ..., x_n$, we wish to make change for a value v using at most k coins; that is, we wish to find a set of $\leq k$ coins whose total value is v. This might not be possible: for instance, if the denominations are 5 and 10 and $k = 6$, then we can make change for 55 but not for 65.

   So, our question is: given $x_1, \ldots x_n$ and a target v, is it possible to make change for v using at most k coins of denominations $x_1, \ldots x_n$?

   (a) Show that this can be formulation as an integer linear programming problem.

   (b) Can we relax this to an LP and still get an integral solution? Either prove it or give a counterexample.

3. In this problem, please assume that algorithms are only allowed to perform pairwise comparisons of the form "is $A_i < A_j$?" or evaluate $f(i)$ at a chosen index i.

   Let $A[1..n]$ be an array of *distinct* real numbers. An index i is called a *local minimum* if:

   - $A[i] < A[i-1]$ (if $i > 1$), and
   - $A[i] < A[i+1]$ (if $i < n$).

   (a) Give a divide-and-conquer algorithm that finds a local minimum using $O(\log n)$ comparisons. [Hint: this should be easy, and you saw it already on a practice problem!]

   (b) Prove a lower bound, in that any algorithm must make $\Omega(\log n)$ comparisons in the worst case.

(c) Now, consider your answers to the above. Is there a "gap", if you consider exact constants instead of big-O or big-Ω only? Justify your answer a bit, although no formal proof is really needed here, just an explanation.