Advanced Algorithms Challenge Problems

LA Math Circle

Problem 1. Write a polynomial time algorithm to solve the longest common subsequence problem, which is the problem of given two lists of numbers L_1 and L_2 to find a list of numbers L_3 which is as long as possible while still being a subsequence of both L_1 and L_2.

We say a list $L_2 = b_1, \ldots, b_m$ is a subsequence of another list $L_1 = a_1, \ldots, a_n$ if there are $1 \leq j_1 < \cdots < j_m \leq n$ such that $b_i = a_{j_i}$. For example, (1,3) is a subsequence of (1,2,3), but (2,1) is not.

Problem 2. There are k monsters waiting to kill you, each with a number of health points h_k and an attack strength a_k. At each point in time, you have the chance to attack one monster to lower its health by one, then the monsters each attack you lowering your health by the sum of the attack strengths of the still alive monsters (i.e.- those whose health is positive).

Find a polynomial time algorithm with input a list H giving the monsters' health points and a list A giving their attack strengths and output the optimal order in which to attack the monsters (optimal meaning a way which minimizes the damage done to you).

Problem 3. Find an algorithm that solves four-peg towers of hanoi in $2^{O(\sqrt{n})}$ steps. Formally, we want it to take as input a number of disks n and to output a list of pairs of numbers $((x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m))$ where (x_i, y_i) represents the action of moving the disk on the x_i^{th} peg to the y_i^{th} peg.

Problem 4. Write a program which finds the number of different ways to write a natural number as the sum of other natural numbers. Two ways are considered distinct only if there is no way to rearrange the addends to get the same sum.

For example, the various ways to rewrite 3 as a sum of other naturals is 3=1+1+1, 3=2+1, 3=3. Note that 3=1+2 doesn't count as distinct since if we flip the 1 and the 2 we get 3=2+1 which is already in our list.