CSCI 3110 Tutorial 7

Reviewed June 28, 2019

I suspect that everybody has played with dominoes before, not the line-them-up-and-tip-them-over game, but the one where the numbers on the dominoes matter. Every domino is a pair of numbers $[x : y]$. Your goal is to form longer and longer sequences of dominoes by attaching new dominoes to either end of your sequence, given the constraint that, if you add domino $[x : y]$ after domino $[v : w]$, then x must equal w. This leads us to the following definition: A *domino sequence* is a sequence

$$([x_1 : y_1], [x_2 : y_2], \dots, [x_n : y_n])$$

such that, for $1 \le i < n$, $y_i = x_{i+1}$. For real-life dominoes, all the x_i and y_i are integers between 1 and 6. To make things interesting, let us allow them to be integers between 1 and n.

Here's the problem: Given a sequence

$$([x_1 : y_1], [x_2 : y_2], \dots, [x_n : y_n])$$

that may or may not be a domino sequence, find the longest subsequence

$$([x_{i_1} : y_{i_1}], [x_{i_2} : y_{i_2}], \dots, [x_{i_m} : y_{i_m}])$$

that is a domino sequence. Indices $i_1, i_2, \dots, i_m$ have to be monotonically increasing, that is, $i_1 < i_2 < \cdots < i_m$; but these indices don't have to be consecutive, that is, it isn't necessarily the case that $i_{j+1} = i_j + 1$, for any j.

a. Develop an algorithm that, given an arbitrary input sequence, S, of n dominoes, finds the longest domino sequence contained in S. Argue that the algorithm is correct and analyze its running time.

b. Using appropriate data structures to maintain information as your algorithm constructs a solution, you will be able to reduce the running time of your algorithm to $\mathrm{O}(n \lg n)$. Explain which data structures are required and how the algorithm can use them to solve the problem in $\mathrm{O}(n \lg n)$ time.

c. Using the same idea, but using a different data structure to store intermediate information, the running time can be reduced to $\mathrm{O}(n)$ in the worst case. Develop such a linear-time algortihm instead of the $\mathrm{O}(n \lg n)$-time algorithm. (Hint: This would not work if the numbers on the dominoes were real numbers.)