

Assignment 9
CSCI 3110: Design and Analysis of Algorithms
Due July 24, 2018

Banner ID: _____

Name: _____

Banner ID: _____

Name: _____

Banner ID: _____

Name: _____

Assignments are due on the due date before class and have to include this cover page. Plagiarism in assignment answers will not be tolerated. By submitting their answers to this assignment, the authors named above declare that its content is their original work and that they did not use any sources for its preparation other than the class notes, the textbook, and ones explicitly acknowledged in the answers. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer and possibly to the Senate Discipline Committee. The penalty for academic dishonesty may range from failing the course to expulsion from the university, in accordance with Dalhousie University's regulations regarding academic integrity.

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. 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 $1 \leq i < n$, $y_i = x_{i+1}$. 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, $i_1 < i_2 < \dots < i_m$, 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 $O(n^2)$ -time 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, reduce the running time of your algorithm to $O(n)$.

Hint: To obtain your algorithm, develop a recurrence that describes the length of the longest domino sequence whose last domino is the i th domino $[x_i : y_i]$ for any $1 \leq i \leq n$.