CS5800: Algorithms Spring 2018
Assignment 3

Out: 31 January 2018 **Due: 11 February 2018, 8:59pm**

Instructions:

- The assignment is due at the time and date specified. Late assignments will not be accepted.
- You must work on all the problems and write the solutions by yourself! Finding solutions to homework problems on the web, or by asking students in and outside the course is strictly prohibited. This would be defeat the purpose of learning by doing the assignment.
- You must submit typed solutions. You may use plain text or a word processor like Microsoft Word or LaTeX for your submissions. You may hand-sketch and scan any diagrams that you support your answer.
- If you are not comfortable with Word or Latex, please solve these problems by hand before devoting time to typing them. Do not waste precious time investigating typesetting up front!

1. (30 points) You are provided a sequence of points $P = p_i(x_i, y_i), 1 \leq i \leq n$. These points were drawn on the screen as a single stroke (i.e. without lifting the pen) by a user in a sketching application (Table 1(a)). The sequence is ordered by how the user drew the stroke.

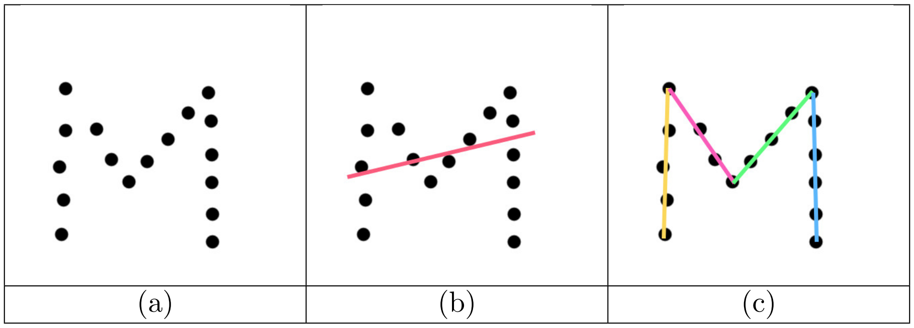
(a) (b) (c)

Table 1: (a) the set of points (b) the result of fitting one least-squares line (c) the result of fitting 4 least-squares line segments

We can fit a line $y = ax + b$ to these set of points using a least-squares formulation (Table 1(b)). In this formulation, the values of a and b can be computed as:

$$a = \frac{n\sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)}{n\sum_i {x_i}^2 - (\sum_i x_i)^2}$$

$$b = \frac{\sum_i y_i - a\sum_i x_i}{n}$$

The error of this fit can be measured as

$e = \sum_i (a * x_i + b - y_i)^2$

The issue is that the drawn points may not form just one line, but a chain of connected line segments (Table 1(c)). However we do not know the points where one line segment ends and the next one begins.

We consider a partition of P to be a set $I = i_1, i_2, i_3, ...i_m$ such that $1 \leq i_1 \leq i_2... \leq i_m \leq n$. This partition creates a poly-line by fitting segments to points $(x_1, y_1)...(x_{i_1}, y_{i_1})$, $(x_{i_1}, y_{i_1})...(x_{i_2}, y_{i_2})$,...,$(x_{i_m}, y_{i_m})...(x_n, y_n)$.

We would like to create as few partitions as possible, while also ensuring that a line fits well (i.e. the error of the fit is minimized). Accordingly, we associate a penalty with the partition. The penalty is the sum of the following:

1. The penalty associated with each segment in the partition, a constant C. (this would discourage excessive partitioning)
2. For each segment, the error value of line fit on the points in that segment

Our problem is to determine a partition of minimum penalty. This would correspond to the "best fit of segments over the stroke".

(a) Prove that this problem exhibits the optimal substructure property. That is, an optimal partition contains optimal partitions for sub-problems.

(b) Devise the problem as a recurrence. Hint: let OPT(i) be the penalty for the optimal partition of points $p_1, ...p_i$.

(c) Write the pseudo-code for a complete bottom-up solution to this problem. The output of your algorithm should be the optimal partition.

(d) Analyze your solution in time and auxiliary space.

Note: the fit line segments may not actually pass through the partitioning points (the figure above is misleading in this regard). In practice, we would create the poly-line by considering the intersection points of adjacent segments.

2. (20 points)

Amit has joined a year-old weight-loss boot camp. This includes weekly weigh-ins, measured in pounds. Obviously the goal for each participant is to lose weight over the year. However to increase healthy competition and promote a sustainable healthy lifestyle, ~~the program awards one special point every time a person weighs less than what they did at an earlier point in the program.~~ the program awards points for a losing streak. For example if Amit's weights are $\{180, 178, 182, 175, 174, 177, 173\}$ then $\{182, 177, 173\}$ is a losing streak. At the end of the boot camp, the person who has the longest weight-losing streak wins a special prize.

We must devise an algorithm that computes the longest losing streak for each participant and declares a winner.

(a) Can this problem be solved using dynamic programming? Justify your answer.

(b) State the technical problem formally (i.e. remove the context and state the computational problem).

(c) Provide an algorithm to solve the general version of this problem (i.e. report the longest losing streak for the winner and the weeks that it occurred in) in $O(mn^2)$ time or better, where m is the number of participants, and the program is n weeks long. Your answer may be in pseudo-code. If it is not, you must state it in a precise sequence of steps. You do not have to prove its running time.