CS 341, Fall 2017 A. Lubiw, B. Ma, E. Zima

ASSIGNMENT 4

DUE: Thursday October 12, 7 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read `http://www.student.cs.uwaterloo.ca/~cs341` for general instructions and policies.

1. [5 marks] **Polynomial Time.** Consider the primality testing algorithm that tests if a given k-bit number n is prime by attempting to divide n by i for $2 \leq i \leq \sqrt{n}$, $i \in N$.

   Using big-oh notation, analyze the run time of this algorithm as a function of input size k. To analyze run-time, do not use the word-RAM model. Instead, assume that you have a division method that runs in time $O(j^2)$ on j-bit integers. (This is called the "bit-complexity model".) Does the algorithm run in polynomial time? Why or why not?

2. [10 marks] **Dynamic Programming to break a paragraph into lines.** This problem is about choosing line breaks to format a paragraph nicely. Suppose that the paragraph has n words of lengths $l_1, l_2, \ldots, l_n$ in that order. Each pair of consecutive words in a line must be separated by a blank, and a blank has length 1. Suppose that the maximum line length is L.

   For example, here are three different formatting choices:

   ```
   I propose to consider the
   question, ‘Can machines think?’
   This should begin with definitions of
   the meaning
   of the terms ‘machine’ and ‘think’.

   I propose to consider the question,
   ‘Can machines think?’ This should
   begin with definitions of the meaning
   of the terms ‘machine’ and ‘think’.

   I  propose to consider  the question,
   ‘Can  machines  think?’  This  should
   begin with definitions of the meaning
   of  the  terms ‘machine’ and ‘think’.
   ```

   The second example chooses the line breaks to make the lines roughly equal in length. The third example adds extra blanks between words to right-justify (so all lines have the same length).

   Give a dynamic programming algorithm to choose line breaks and add extra blanks so that all lines have the same length L. A line cannot start or end with a blank. This means that a line cannot have just one word unless that word has exactly L characters, so there are inputs that

have no solution. Let m be the maximum number of blanks in a row that you add between words. Your goal is to find a solution that minimizes m.

Note: If words $i, i+1, \ldots, j$ go on a line then the number of characters in the line is at least $c_{i,j} = j - i + \sum_{k=i}^{j} l_k$. To get line length L, the number of extra blanks you must add is $L - c_{i,j}$, and you will distribute these blanks as equally as possible in the $j - i$ spaces between the words.

Give a dynamic programming algorithm to find the minimum value m. The run-time should be $O(n^2)$. Clearly indicate what your subproblems are, and the order in which you solve them. Justify correctness of your algorithm, and analyze its running time.

3. [10 marks] **Dynamic Programming to break a sequence into harmonious parts.** Suppose you have a sequence of musical notes $s_1 s_2 \ldots s_n$. Some subsequences are said to be *harmonious*. You can test if a sequence is *harmonious* via a unit-time test $H(i, j)$ that returns 1 if the subsequence $s_i s_{i+1} \ldots s_j$ is harmonious, and 0 otherwise. Your goal is to break the input sequence into harmonious subsequences if possible.

   For example, if AB, B, and BAB are the only harmonious sequences, then the sequence $ABBAB$ can be broken into harmonious sequences as $AB \mid B \mid AB$ or as $AB \mid BAB$, but the sequence $ABABA$ cannot be broken into harmonious sequences.

   Give a dynamic programming algorithm to test if a sequence can be broken into harmonious subsequences. The run-time should be ~~$O(n^3)$~~ $O(n^2)$. Clearly indicate what your subproblems are, and the order in which you solve them. Justify correctness of your algorithm, and analyze its running time.