**CPSC 320: Intermediate Algorithm Design** 2019/20 Term 2

Tutorial 8

*Prof. Nick Harvey* *University of British Columbia*

# 1 Word Segmentation

Internet domain names often consist of multiple words concatenated together without spaces, like `vancouversun.com`. Humans reading these URLs must try to infer likely boundaries between consecutive words in the text. This sometimes leads to amusing examples: should `oldmanshaven.com` be interpreted as "Old Man's Haven" or "Old Man Shaven"? There are other example, like `choosespain.com`, or `childrenswear.co.uk`. You may wish to entertain yourself by looking up other examples.

As an algorithm designer, we may wish to understand how to automatically determine these likely boundaries. We call this the Word Segmentation problem.

A simple, reasonably effective approach is to find a segmentation that maximizes the cumulative "quality" of its individual constituent words. Thus, suppose you are given a black box that for any string of one or more letters $X[1..k]$, $k \geq 1$, will return a number $\mathrm{Q}(X)$. This number can be either positive or negative; larger numbers correspond to more plausible English words. So Q(me) would be positive, while Q(ght) would be negative.

Given a long string of letters $Y[1..n]$, a segmentation of $Y$ is a partition of its letters into contiguous nonempty blocks of letters; each block corresponds to a word in the segmentation. The total quality of a segmentation is determined by adding up the qualities of each of its blocks. So we would get the right answer for `choosespain` provided that Q(choose) + Q(spain) was greater than the total quality of any other segmentation of the string.

1. Give a recurrence relation for Max-TQ($i$): the Maximum Total Quality of any segmentation of a string of letters $Y[1..i]$ for $i \geq 0$. Suppose that the empty string has quality zero.

2. Use the recurrence above, design an algorithm to compute Max-TQ using **memoization**.

   The input is a string Y[1..n]. Assume that a function for computing Q($X$) is available, where $X$ is a string of one or more characters.

   What is the runtime of your algorithm?

3. Suppose we have already executed our algorithm that computes Max-TQ, and returns the memoization data structure Soln. Describe another algorithm that receives as input the string $Y[1..n]$ and the array Soln. It must output the segmentation of $Y$ into words of highest total quality. It should run in $O(n^2)$ time.

   **Keener kwestion:** Can you make it run in $O(n)$ time?