15-210 Assignment 03

Karan Sikka
ksikka@cmu.edu
Section C

September 23, 2012

1: Task 2.2

The work of this function is $O(n)$, because the most work-expensive operation called is `map` on input of length n, using a constant work function. `map` is also called on each start of a token, with a function that does work proportional to the length of the token it started on. The total work of these two maps is $O(n)$.

The span of this function is O(log(n) + —l—). The log term comes from performing filter on an input length of n, and the —l— term comes from performing `String.substring` on the longest token.

2: Task 4.2

I would use an inclusive scan to do prefix sums as the first step. Then, I'd scale the real number from [0,1] to [0,sum of frequencies] by multiplying by the sum of all the frequencies in the histogram. Call this number the "threshold". Then I would filter to keep the elements which have a frequency greater or equal to the threshold. Then I would take the first element from the filtered Sequence.

3: Task 4.3

Use an inclusive scan to do prefix sums as in the first step. Store the result in sequence of tuples of type `string * int. Then, create a binary search tree where the keys are the frequencies and the values are the strings.`

`Then, when the function is called, scale the real number by the sum of the frequencies, and call this the frequency threshold. Then traverse the binary tree, but while at each node, keep track of the key where (frequency threshold - key) is the least positive number, and the value the key is associated with.. and the value it is associated with. If the tree balanced, the traversal will take` $O(\log(n))$ `work, and since this is sequential, it will also take` $O(\log(n))$ `span. At the end of the traversal, return the value you have cached.`