6.851: Advanced Data Structures, Spring 2021

Prof. Erik Demaine, Josh Brunner, Dylan Hendrickson, Yevhenii Diomidov

Problem Set 8

Due: Thursday, April 15, 2021

Problem 8.1 [Signature Compression]. Recall that the first step in Signature Sort, as described in Lecture 14, was to reduce each $w = \lg^{2+\varepsilon} n$ bit input word to an $O(\lg^{1+\varepsilon} n)$-bit ***signature***. We achieved this reduction by dividing each word into $k = \lg^{\varepsilon} n$ ***chunks*** of $w/k = \lg^2 n$ bits each and applying a hash function to each chunk, reducing each chunk to $\lg n$ bits. After hashing, our word was of the form

$$0^{w/k-\lg n}\, h_1\, 0^{w/k-\lg n}\, h_2\, \cdots\, 0^{w/k-\lg n}\, h_k,$$

where each chunk hash h_i is $\lg n$ bits long.

Describe an algorithm to **compress** this word so that all of the hashed chunks are shifted maximally to the right and in the same order, so that it has the form needed by the algorithm:

$$0^{w-k\lg n}\, h_1\, h_2\, \cdots\, h_k.$$

Your algorithm should take $O(1)$ time using the word RAM operations `+`, `-`, `*`, `/`, `%`, `&`, `|`, `~`, `^`, `<<`, and `>>`. You may assume that $\varepsilon < 1$.