

6.851 ADVANCED DATA STRUCTURES (SPRING'10)

Prof. Erik Demaine Dr. André Schulz TA: Aleksandar Zlateski

Problem 7 *Due: Thursday, Apr. 1*

Be sure to read the instructions on the assignments section of the class web page.

Finding the most significant 1 bit. Several times in lecture, we've needed the operation of finding the bit position (index) of the most significant 1 bit in a word x . This is equivalent to computing $\lfloor \lg x \rfloor$. In this problem, you'll solve this problem in constant time using a word RAM with standard C operations on integers ($+$, $-$, $*$, $/$, $\%$, $\&$, $|$, \sim , \wedge , \ll , \gg).

(a) Suppose we divide a w -bit word into \sqrt{w} chunks, each $b = \sqrt{w}$ bits long. Describe how to compute in $O(1)$ time a word that replaces each chunk with either 0^b if the chunk is all 0s, or 10^{b-1} if the chunk has a 1.¹

(b) Prove that you can compress the chunk summary computed in part (a) down to b consecutive bits, preserving their order. (Hint: multiply, mask, shift.)

(c) Describe how to compute the most significant 1 bit in the chunk summary word computed in part (b). (Hint: Use a static fusion-tree node. But take care not to rely on finding the most significant 1 bit.)

(d) Describe how to compute the most significant 1 bit in the most significant chunk with a 1 in it, and thus compute the overall most significant 1 bit.

¹Here 0^k denotes 0 repeated k times.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.851 Advanced Data Structures
Spring 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.