Homework 9

Due 10/05/16

September 30, 2016

1. Describe a divide-and-conquer algorithm that accepts a positive integer n and computes $\lfloor \lg n \rfloor$ (that is, the largest integer x such that $2^x \leq n$). Your algorithm should take $O(\lg(\lg n))$ time.

   Hint: you may wish to base your approach on one-sided binary search, which starts at 1, doubles the value until it becomes too large, then performs binary search between the last value that worked and the first value that failed. (Divide-and-conquer algorithms aren't required to be recursive.) You may assume that the square root function takes $\Theta(1)$ time, though there is an $O(\lg \lg n)$ algorithm that does not use the square root.