Homework 2. ORIE 6125.

Saul Toscano-Palmerin

February 21, 2018

In this homework you will implement red-black trees, sorting algorithms and the Bellman-Ford shortest path algorithm. You have to prove the time and space complexity of your algorithms. You should follow the rules for writing quality code and test at least 85% of your code (see this template). The homework deadline is March 7.

1. Implement red-black trees as a class.

2. Suppose that we have an array with 'a's and 'z's. Write a program that finds the subarray of maximum size with the same number of 'a's and 'z's. Expected time complexity must be $O(n)$. Example:

   (a) Input: ['a', 'a', 'z', 'z', 'a']. Output: 0 to 3 Or 1 to 4

3. Suppose that we have a sorted array $[0, .., n-1]$. We choose an element i of the array creating two subarrays: $[0, .., i-1]$ and $[i+1, .., n-1]$. We interchange these subarrays, which gives the new array: $[i+1, .., n-1, i, 0, .., i-1]$. Code a function that finds an element in the modified array in $O(\log n)$ time. Example:

   (a) Input: [15, 16, 17, 18, 19, 20, 11,12, 13], key= 13. Output: Index 8

4. Implement quick sort but, when a vector has a size smaller than K in a call, use insertion sort instead. Find the value of K that maximizes empirical performance. (Note that K = 1 is the same as not having a K at all.) The space complexity of quick sort must be $O(\log n)$, and you should use the median of the first, middle and last element as a pivot.

5. Use the Bellman-Ford shortest path algorithm to code a function that finds negative cycles in a graph. The time complexity should be $O(nm)$, and the space complexity should be $O(n)$ where m is the number of edges, and n is the number of nodes. You should prove that your algorithm finds negative cycles.