Advanced Algorithms vEB trees, orthogonal range search Teaching Block 2
COMS31900 Spring 2019

Problem sheet 2

Please feel free to discuss these problems on the unit discussion board or directly with your colleagues. If you would like to have your answers marked, please either hand them in in person at a lecture or problems class. Submitted work will be marked as quickly as possible, ideally within one week of being handed in.

1. The dynamic predecessor problem can be defined to be the dynamic dictionary problem with the addition of the PREDECESSOR operation. Recall that given a s :et of integers Y , the PREDECESSOR of x (which may not be in Y) is the largest $v \in Y$ such that $v \leq x$. The dynamic predecessor problem could solved by either a self-balancing search tree (such as an AVL or Red-Black tree) or a van Emde Boas tree.
   (a) Give one advantage of using a van Emde Boas tree over a self-balancing search tree.
   (b) Give one advantage of using a self-balancing search tree over a van Emde Boas tree.
2. How can you perform DELETE in a van Emde Boas tree ? You should write your solution in the same style as in the lecture slides. What is the relevant recurrence relation for the time complexity of DELETE and what is its solution in terms of big O complexity ?
3. Looking at slide 189 of the van Emde Boas tree lectures slides, consider a version of van Emde Boas trees where the new minimum is always recursively inserted into the tree instead of being stored at only one level. Write down the recurrence relation for the time complexity of the ADD operation and give its solution in big O notation.
4. In 2D orthogonal range search, justify why we the number of 1D lookups performed is $O(\log n)$ (see slide 165).
5. In some applications one is interested only in the number of points that lie in a range rather than in reporting all of them. Such queries are often referred to as range counting queries. In this case one would like to avoid having an additive term of $O(k)$ in the query time.
   (a) Describe how a 1-dimensional range tree can be adapted such that a range counting query can be performed in $O(\log n)$ time. Prove the query time bound.
   (b) Using the solution to the 1-dimensional problem, describe how 2-dimensional range counting queries can be answered in $O\left(\log^2 n\right)$ time. Prove the query time.
   (c) (*)Describe how fractional cascading can be used to improve the running time by a factor of $O(\log n)$ for 2-dimensional range counting queries.