Advanced Algorithms - Problem Sheet 2

Dom Hutchinson

March 1, 2020

Question - 1.

The dynamic predecessor problem can be defined to be the dynamic dictionary problem with the addition of the `PREDECESSOR` operation. Recall that gien a set 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 be solved by either a self-balancing search tree (such as an AVL or Red-Black Tree) or a van Emde Boas tree.

Question 1 a) - Give one advantage of using a van Emde Boas Tree Over a self-balancing search tree.

Answer 1 a) Do not need to reshape the tree each time an insertion or deletion occurs.

Question 1 b) - Give on advantage of fusing a self-balancing search tree over a van Emde Boas Tree.

Answer 1 b) Run time for operations depends on the size of set being stored, not on the size of the universe the set is taken from. Thus, if the set is much smaller than the universe then a self-balancing tree should be quicker.

Question - 2.

How can you perform `DELETE` in a van Emde Boas tree?
What is the relevant recurrence relation for the time complexity of `DELETE` and what is its solution in terms of big O complexity?

Answer 2`DELETE`(x)

1. Determine which $B[i]$ the element x belongs in.
2. If $B[i]$ is empty then stop.
3. Else If $B[i]$ is root set $B[i][x] = 0$ (adjusted for offset)
4. Else delete x from $B[i]$ adjusted for offset.
5. If $B[i]$ is empty set $C[i] = 0$.

$O(\log \log u + \sqrt{u})$?????

Question - 3.

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.

Answer 3