CSCI 5610 Solutions to Exercises(List 1)

Anonymous

January 30, 2020

Lecture 2: Problem 1

1. For the $(-\infty, y)$ case:

   1. Initially, $\Sigma = \Phi$, and set u to the root of T .
   2. If the key of u equals x, then add $slab(lc(u)))$ to Σ, and stop. (lc: left child,rc: right child)
   3. If the key of $u < x$, then add $slab(lc(u)))$ to Σ, and let $u = rc(u)$.
   4. If the key of $u > x$, let $u = lc(u)$, repeat from 2.

2. For the $[x, y)$ case, we calculate canonical slabs of $[x, \infty]$ and $(-\infty, y)$. Then we remove the two unbounded intervals containing ∞ and $-\infty$.

Lecture 2: Problem 2

Construct a BST on the age of people with each node u containing the following additional information:(1) the range of ages covered in the subtree rooted on u, namely l, r; (2) the maximum salary in the subtree rooted on u.

For a query $[x, y]$, we will handle it in a recursive way:

Algorithm 1 queryMax(u,x,y)

if $y < u.l$ **or** $x > u.r$ **then**
    return $-\infty$
end if
if $x \leq u.l$ **and** $y \geq u.r$ **then**
    return u.maxSalary
end if
if $(x \geq u.l$ **and** $x \leq u.r)$ **or** $(y \geq u.l$ **and** $y \leq u.r)$ **then**
    return max(queryMax(lc(u), x, y),queryMax(rc(u), x, y)))
end if

Lecture 2: Problem 2 Con't

As for the query time of O(log n), we can see that no more than 4 nodes will be visited at each level and there are O(log n) levels in total. (Click here for reference)

Lecture 2: Problem 3

In each node u, we store the size of the subtrees rooted on lc(u) and rc(u) as l_cnt and r_cnt correspondingly. In order to find the k-th largest element in S, we define the following function:

Algorithm 2 find_kth_largest(u,k)

if $u.l\_cnt \geq k$ **then**
    return find_kth_largest(lc(u),k)
end if
if $u.l\_cnt + 1 == k$ **then**
    return u.key
end if
if $u.l\_cnt + 1 \leq k$ **then**
    return find_kth_largest(rc(u),k-(u.l_cnt + 1))
end if

Lecture 2: Problem 4

To get a 2-3 tree on the set S\[x,y], we can use the existing split and join functions.

1. Use x to split S into S_1 and S_2. All the values in S_1 is smaller than x. We get a 2-3 tree on S_1 and a 2-3 tree on S_2.
2. Use y to split S_2 into S_{21} and S_{22} in the same way.
3. Delete y in the 2-3 tree on S_{22} if it exists.
4. Join S_1 and S_{22} to get a 2-3 tree on set S\[x,y]

Lecture 2: Problem 5*(optional)