Handin 5

Thomas Vinther & Jens Kristian Refsgaard Nielsen

21-09-18

1 Radix trees

Given a set $S = \{s_j | j = 1 \dots m : s_j$ bit string of length $n_j\}$, denote by n the sum of the lengths of the elements from S i.e.

$$\sum_{j=1}^{m} n_j = n \tag{1.1}$$

. We wish to sort the elements of S by using the standard lexicographical sort, defined by:

$$a_0 a_1 \dots a_p \leq b_0 b_1 \dots b_q \iff \tag{1.2}$$
$$\exists j : 0 \leq j \leq \min(p, q) : a_i = b_i \text{ for } i = 0 \dots j-1 \text{ and } a_j < b_j \tag{1.3}$$
$$p < q \text{ and } a_i = b_i \text{ for } i = 0 \dots p \tag{1.4}$$

1.1 Building the tree

We construct the tree using the following algorithms.

Time	Line nr	Pseudocode
	0	Build-Radix-Tree(S)
1	1	T.root.key = false
m	2	for $s_j \in$ S
$n_j + 1$	3	InsertRT(s_j,T,T.root)
	0	InsertRT($a_0 a_1 \dots a_p$,T,x)
1	1	if $a_0 = 0$
1	2	if x.left = nil
1	3	x.left.key = false
p+1	4	InsertRT($a_1 \dots a_p$,T,x.left)
1	5	if $a_0 = 1$
1	6	if x.right = nil
1	7	x.right.key = false
p+1	8	InsertRT($a_1 \dots a_p$,T,x.left)
1	9	if $a_0 =$ nil
1	10	x.key = true

Correctness: In accordance with the example in the book we take an element of our set S and finds its correct position by going left if the cipher is a 0 or right if it is a 1, and recursively calling the function without the "first"cipher. In this manner we will clearly reach the correct position of our element. Along the path we ensure that all the nodes are instanciated with the

lines 2,3,6 and 7. When we run out of ciphers the algorithm puts a true value in the node and stops.
Time: Note that for $s_j \in S$ the length n_j is equal to the height of s_j in our Radix tree. Consider firstly InsertRT($s_j, T, T.root$) For each passed node we make up to 4 calculations, so we have $T(s_j) \leq c_u n_j$ but on the other hand we always go all the way down, with no reuse of previous work, so $c_l n_j \leq T(s_j)$ for some small constants $c_l \leq c_u$, in total $T(s_j) = \Theta(n_j)$. Now for Build-Radix-Tree(S) we get

$$T(S) = \sum_{j=1}^{m} T(s_j) = \sum_{j=1}^{m} \Theta(n_j) = \Theta\left(\sum_{j=1}^{m} n_j\right) \overset{(1.1)}{=} \Theta(n) \tag{1.5}$$

As wanted.

1.2 Sorting the tree

To print our radix tree in sorted order we use a modified preorder treewalk algorithm.

Time	Line nr	Pseudocode
	0	preorderRTwalk(x,$a_0 a_1 \dots a_k$)
1	1	if $x \neq$ nil
1	2	if x.key
1	3	print $a_0 a_1 \dots a_k$
1	4	preorderRTwalk(x.left . $a_0 a_1 \dots a_k 0$)
1	5	preorderRTwalk(x.right , $a_0 a_1 \dots a_k 1$)

Correctness: