Data Structures and Algorithms

COSC 336 Assignment 7 - Solutions

Exercise 1. Show similarly to Fig 8.3 on page 198 in the textbook, how RadixSort sorts the following arrays:

1. 34, 9134, 20134, 29134, 4, 134

2. 4, 34, 134, 9134, 20134, 29134

3. 29134, 20134, 9134, 134, 34, 4

1. 34, 9134, 20134, 29134, 4, 134
(Zeroes are added to make all numbers same length. Longest number has 5 digits, so zeroes are added to other nums. accordingly)

00034		00004		00004		00004		00004
09134		00034		00034		00034		00034
20134	→	09134	→	09134	→	20134	→	00134
29134		20134		20134		00134		09134
00004		29134		29134		09134		20134
00134		00134		00134		29134		29134

2. 4, 34, 134, 9134, 20134, 29134

00004		00004		00004		00004		00004
00034		00034		00034		00034		00034
00134	→	00134	→	00134	→	00134	→	00134
09134		09134		09134		20134		09134
20134		20134		20134		09134		20134
29134		29134		29134		29134		29134

3. 29134, 20134, 9134, 134, 34, 4

29134		00004		00004		00004		00004
20134		29134		00034		00034		00034
29134	→	20134	→	29134	→	20134	→	00134
00134		09134		20134		00134		09134
00034		00134		09134		29134		20134
00004		00034		00134		09134		29134

Exercise 2. Present an $O(n)$ algorithm that sorts n positive integer numbers $a_1, a_2, \ldots, a_n$ which are known to be bounded by $n^2 - 1$ (so $0 \le a_i \le n^2 - 1$, for every $i = 1, \ldots, n$. Use the idea of Radix Sort (discussed in class and presented in Section 8.3 in the textbook).

Note that in order to obtain $O(n)$ you have to do Radix Sort by writing the numbers in a suitable base. Recall that the runtime of Radix Sort is $O(d(n+k))$, where d is the number of digits, and k is the base, so that the number of digits in the base is also k. The idea is to represent each number in a base k chosen so that each number in $\{0, 1, \ldots, n^2 - 1\}$ requires only 2 "digits," so $d = 2$. Explain what is the base that you choose and how the digits of each number are calculated, in other words how you convert from base 10 to the base. Note that you cannot use the base 10 representation, because $n^2 - 1$ (which is the largest possible value) requires $\log_{10}(n^2 - 1)$ digits in base 10, which is obviously not constant and therefore you would not obtain an $O(n)$-time algorithm. By the same argument we see that no base k that is constant works, therefore k has to depend on n. In your explanations you need to indicate the formula that gives k as a function of n, and show that $d = 2$ "digits" are enough to represent all the numbers in the range $\{0, 1, \ldots, n^2 - 1\}$.

Illustrate your algorithm by showing on paper similar to Fig. 8.3, page 198 in the textbook (make sure you indicate clearly the columns) how the algorithm sorts the following 2 sequences:

(a) 45, 98, 3, 82, 132, 71, 72, 143, 91, 28, 7, 45.

In this example $n = 12$, because there are 12 positive numbers in the sequence bounded by $143 = 12^2 - 1$.

(b) 45, 98, 3, 82, 132, 71, 72, 143, 91, 28, 7, 45, 151, 175, 145, 399, 21, 267, 346, 292.

In this 2-nd example $n = 20$, because there are 20 positive numbers in the sequence bounded by $399 = 20^2 - 1$.

Note: if you use a base b bigger than 10, you do not need to invent symbols for the digits larger than 10; instead use as digits the numbers $0, 1, \ldots, b - 1$ represented in base 10. For instance if you use base, say 25, the digits will be: 0,1, ..., 9, 10, 11, ..., 23, 24. So we view '10', '11', etc., as a single symbol. For instance in this representation the number 9 23 written in base 25 has 2 digits: 9 and 23.

Solution: If we use some base b, the smallest number with 2 digits is of course

0 0,

which is equal to the value 0, and the largest is

b-1 b-1,

which represents the number $(b-1)b + (b-1) = b^2 - 1$.

Therefore to represent the numbers in the set $\{0, 1, \ldots, n^2 - 1\}$, we can use base n.

The 2 examples:

(a) 45, 98, 3, 82, 132, 71, 72, 143, 91, 28, 7, 45.

There are 12 numbers so we write them in base 12. Written in base 12, the numbers need only 2 digits and they are in order:

3 9 8 6 3 6 10 11 0 5 11 6 0 11 11 7 7 2 4 7 3 9

(For instance $143 = 11 \times 12 + 11$, so 143 in base 12 is 11 11 .)

Next we do Radix Sort

sort last col.	sort first col.	convert to base 10
11 0	0 3	3
6 0	0 7	7
8 2	2 4	28
0 3	3 9	45
2 4	3 9	45
7 7	5 11	71
0 7	6 0	72
3 9	6 10	82
3 9	7 7	91
6 10	8 2	98
5 11	11 0	132
11 11	11 11	143

(b) 45, 98, 3, 82, 132, 71, 72, 143, 91, 28, 7, 45, 151, 175, 145, 399, 21, 267, 346, 292.

Here there are 20 numbers, so we represent them in base 20. Written in base 20, the numbers need only 2 digits and they are:

2 5   4 17   0 3   4 2   6 12   3 11   3 12   7 3   4 11   1 8   0 7   2 5   7 11
8 14   7 5   18 18   1 1   12 7   16 6   13 12

Next we can do Radix Sort using d=2, because there are 2 columns.

sort last col.	sort first col.	convert to base 10
1 1	0 3	3
4 2	0 7	7
0 3	1 1	21
7 3	1 8	28
2 5	2 5	45
2 5	2 5	45
7 5	3 11	71
H 6	3 C	72
0 7	4 2	82
13 7	4 11	91
1 8	4 18	98
3 11	6 12	132
4 11	7 3	143
7 11	7 5	145
6 12	7 11	151
3 12	8 15	175
14 12	13 7	267
8 15	14 12	292
4 18	17 6	346
18 18	18 18	399

$\square$