

CMPT 280

Topic 24: Efficient Sorting Algorithms

Mark G. Eramian

University of Saskatchewan

References

- Textbook, Chapter 24

Linear Sorts

- Sorting by comparison of elements is $\Omega(n \log n)$.
- If we use other approaches, we can sort in linear time, but the linear time sorts we present are not general-purpose; e.g. MSD Radix sort does not sort numbers in increasing order.

Exercise 1

Bucket Sort

```
1 // Sort sequence S using element keys. Assume element keys
2 // are between 0 and d-1.
3 Algorithm bucketSort(S)
4 S - sequence to be sorted
5
6 let B be an array of d sequences, each initially empty
7
8 for each item x in S
9     let k be the key of x
10    remove x from S and append it to sequence B[k].
11
12 // S is now empty
13
14 for i = 0 to d - 1
15     for each item x in sequence B[i]
16         remove x from B[i] and add it to the end of S.
```

- What are the limitations of bucket sort?
- What is its time complexity? Space complexity?
- Trace the algorithm when sorting the sequence:
9, 1, 2, 6, 8, 6, 2, 7, 1, 9.

Exercise 2

Radix Sorting of Integers

- a) What would be the output of MSD radix sort for the input array:

0	1	2	3	4	5	6	7
916	42	83	7	95	28	265	614

- b) What would be the output of LSD radix sort for the input array from part a)?

Exercise 3

Radix Sorting of Strings

- a) What would be the output of MSD radix sort for the input array:

0	1	2	3	4	5	6	7
"winter"	"is"	"coming"	"you"	"win"	"or"	"you"	"die"

- b) What would be the output of LSD radix sort for the input array in part a)?

Exercise 4

- Trace MSD radix sort for the input array:

0	1	2	3	4	5	6	7	8	9	10
916	42	83	95	28	265	614	262	263	951	911

Exercise 4

```
1  Algorithm MsdRadixSort(keys, R)
2  keys - keys to be sorted
3  R - the radix
4
5  sortByDigit(keys, R, 0)
6
7
8  Algorithm sortByDigit(keys, R, i)
9  keys - keys to be sorted
10 R - the radix
11 i - digit on which to partition -- i = 0 is the left-most digit
12
13   for k = 0 to R-1
14     list[k] = new list // Make a new list for each digit
15
16   for each key
17     if the i-th digit of the key has value k add the key to list k
18
19   for k = 0 to R-1
20     if there is another digit to consider
21       if list[k] is small
22         use an insertion sort to sort the items in list[k]
23       else
24         sortByDigit(list[k], i+1)
25
26   keys = new list // empty the input list
27
28   For k = 0 to R-1
29     keys = keys append list[k]
```


Exercise 5

- Trace LSD radix sort for the input array:

0	1	2	3	4	5	6	7	8	9	10
916	42	83	95	28	265	614	262	263	951	911

Exercise 5

```
1  Algorithm LsdRadixSort(keys)
2  keys - array of keys to be sorted
3
4  For each digit d from least significant to most significant
5
6      /* keys are already sorted by digits d+1, d+2, ...
7       * so now use a stable sort to sort by digit d */
8
9      for k = 0 to R-1 // for each possible value of digit d
10         list[k] = new list
11
12     for each key in order from first to last
13         if the d-th digit of the key has value k
14             add the key to the end of list[k]
15
16     keys = new list // Empty the list 'keys'
17
18     for k = 0 to R-1
19         keys = keys append list[k]
```