

CSC 301: Analysis of Algorithms

Lecture 17: Lower Bounds, Sorting in Linear Time

William Rebelsky
[rebelskyw]

Grinnell College

Wednesday 28th February, 2024

Reminders

- Homework 3 grades will be released soon. A 4 is required for an S
- Homework 4 is due Tonight
- Homework 5 has been released on gradescope, it is due in two weeks. I recommend you look at it tonight and try to get most of it done by next Wednesday
- Quiz 4 grades have been released, a 1 is required for an S. If you did not get a 1, the redo quiz on Friday will allow you to upload a file with your work instead of being all or nothing
- Today's Lecture will be the last material on the Midterm

Table of Contents

1 Counting Sort

2 Radix Sort

3 Bucket Sort

4 Practice

Counting Sort Basics

- Assumes that each of the n inputs is an integer in the range 0 to k
- Runs in $\Theta(k + n)$ time
- If k is $O(n)$ then it runs in $\Theta(n)$ time

Counting Sort Description

- Initialize two arrays $B[1:n]$ and $C[0:k]$
- Loop through the elements in A , increment the corresponding index of C
- Loop through the elements in C , add the prior value (to get the ending index where the value occurs)
- Loop through the elements of A from n down to 1.
 $B[C[A[j]]] = A[j]$ (place it in the corresponding index)
 $C[A[j]] = C[A[j]] - 1$ (reduce final index to account for duplicates)

Counting Sort Practice

- Write Pseudocode for Counting Sort
- Prove that it is $\Theta(n)$
- Why is the last Loop important? If we don't care about stability, is there a different way you could implement this idea?

Table of Contents

1 Counting Sort

2 Radix Sort

3 Bucket Sort

4 Practice

Radix Sort Basics

- Used by the old card-sorting machines
- 80 columns, 12 places a hole can be per column
- Put the card into one of 12 bins depending on which row has a hole in the column of interest
- In general, repeated stable sorts from least to most significant digit

Radix Sort Description

- Radix-Sort(A, n, d) Array A with n values and d digits
- for i in 1 to d , use a stable sort to sort the array $A[1:n]$ on digit i
- Lemma 8.3: Radix-sort works in $\Theta(d(n + k))$ if the stable sort takes $\Theta(n + k)$
- Lemma 8.4: Given n b -bit numbers and a positive integer $r \leq b$, Radix-Sort works in $\Theta((b/r)(n + 2^r))$

Radix Sort Practice

- Prove Lemma 8.3

Table of Contents

1 Counting Sort

2 Radix Sort

3 Bucket Sort

4 Practice

Bucket Sort Basics

- Assumes input is drawn from uniform distribution $[0, 1)$
- Average case $O(n)$

Bucket Sort Description

- Divide the interval $[0, 1)$ into n equal sized buckets
- Place each number in it's bucket
- Since uniform distribution, expect few collisions
- Sort each bucket using insertion sort
- Concatenate the lists
- The Expected value of the Insertion sort is $2 - 1/n$ per bucket. See pages 184-185 in the online textbook

Bucket Sort Practice

- Convince your group that The algorithm works

Table of Contents

1 Counting Sort

2 Radix Sort

3 Bucket Sort

4 Practice

Practice 1

- 7.1-1 Illustrate the operation of Partition on the Array [9,5,12,8,7,4,21,2,6,11]
- 8.2-1 Illustrate the operation of Counting-Sort on the Array [6,0,2,0,1,3,4,6,1,3,2]
- 8.3-1 Illustrate the operation of Radix-Sort on the Array [Cow,Dog,Sea,Rug,Row,Mob,Box,Tab,Bar,Ear,Tar,Dig,Big,Tea,No]
- 8.4-1 Illustrate the operation of Bucket-Sort on the Array [.79,.13,.16,.64,.39,.2,.89,.53,.71,.42]
- Hint for Homework 5: Write your results down on paper not a whiteboard

Practice 2

- 8.3-3 Prove that Radix Sort Works
- 7.2-4 Consider an almost sorted array. Why would sorting it with Insertion sort be better than sorting it with Quicksort?
- 8.4-3 Let X be a random variable equal to the number of heads in two flips of a fair coin. What is $E[X^2]$ and What is $E^2[X]$?
- 8-5 Average sorting. Suppose that we just require that elements increase on average. An Array is k -sorted if
$$\frac{\sum_{j=i}^{i+k-1} A[j]}{k} \leq \frac{\sum_{j=i+1}^{i+k} A[j]}{k}$$

Practice 3

- 8-5 Average sorting. Suppose that we just require that elements increase on average. An Array is k-sorted if

$$\frac{\sum_{j=i}^{i+k-1} A[j]}{k} \leq \frac{\sum_{j=i+1}^{i+k} A[j]}{k}$$

- What does it mean to be 1-sorted
- Give a permutation of the numbers 1:10 that is 2-sorted but not sorted
- Prove that an n-element array is k-sorted if and only if $A[i] \leq A[i + k]$ for all $i \leq n - k$