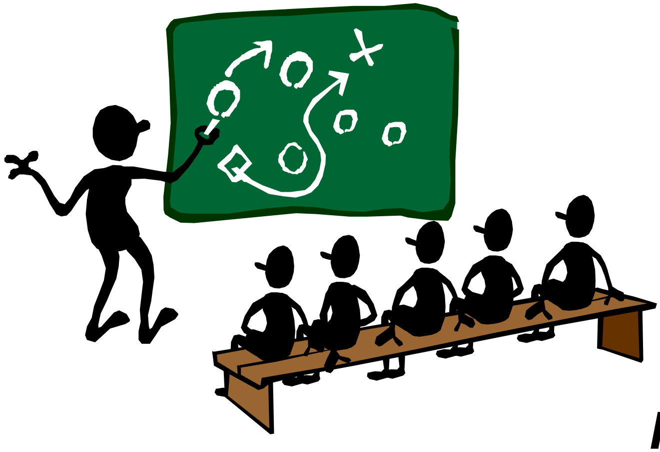
Algorithms – Chapter 8
Sorting in Linear Time



Juinn-Dar Huang

Professor

jdhuang@mail.nctu.edu.tw

August 2007

Rev. '08, '11, '12, '15, '16, '18, '19, '20, '21

Counting Sort

Assume that each of the n input elements is an integer in the range 0 to k for some integer k

COUNTING_SORT(A,B,k)

1 **for** $i \leftarrow 0$ **to** k

2 **do** $c[i] \leftarrow 0$

3 **for** $j \leftarrow 1$ **to** length[A]

4 **do** $c[A[j]] \leftarrow c[A[j]]+1$

5 ▶ $c[i]$ now contains the number of elements equal to i

6 **for** $i \leftarrow 1$ **to** k

7 **do** $c[i] \leftarrow c[i]+c[i-1]$

8 ▶ $c[i]$ now contains the number of elements less than or equal to i

9 **for** $j \leftarrow length[A]$ **downto** 1

10 **do** $B[c[A[j]]] \leftarrow A[j]$

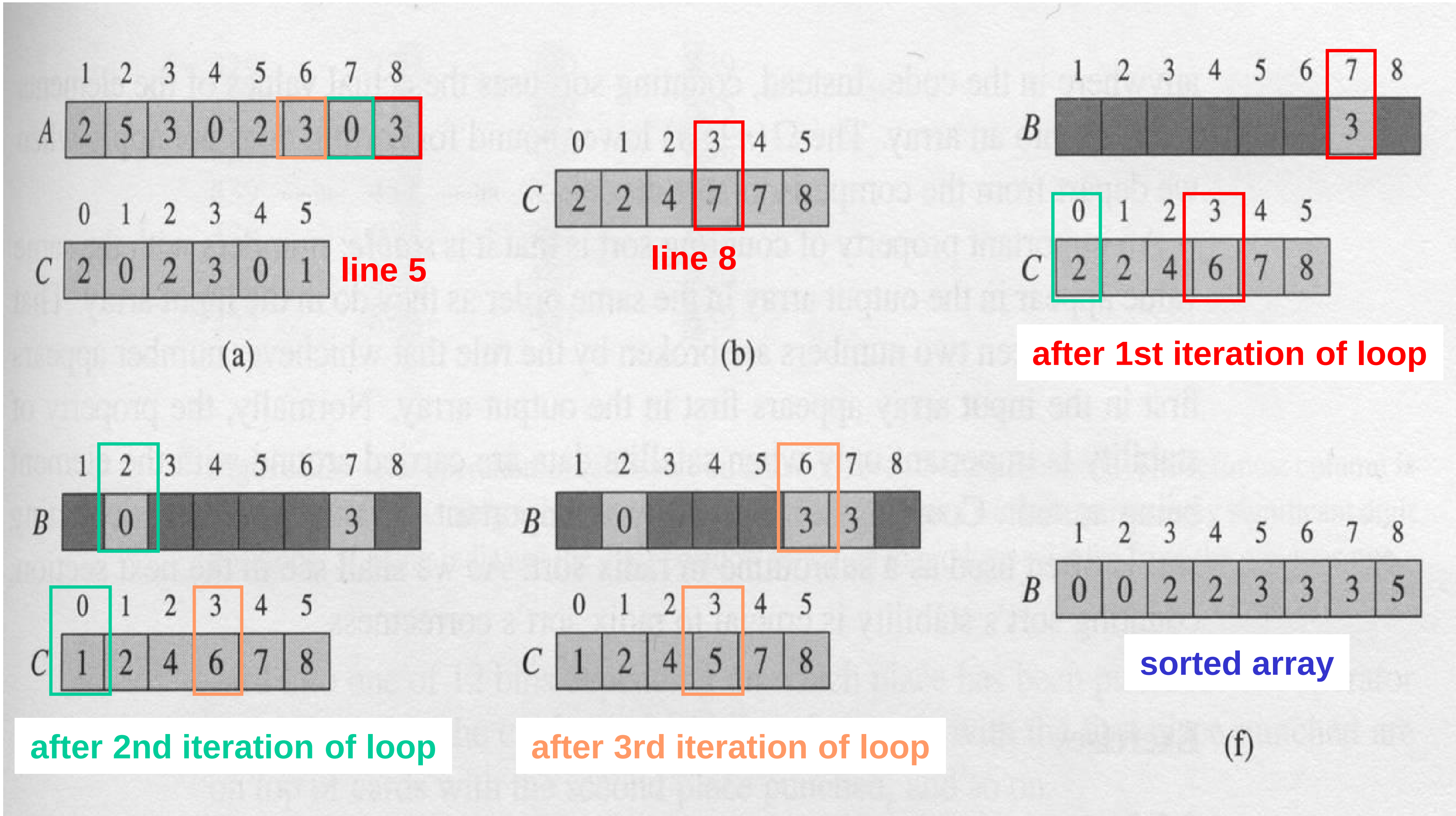
11 $c[A[j]] \leftarrow c[A[j]]-1$

Sort Array A into Array B

Sorting in Linear Time

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Example



Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Time Complexity

- Time complexity
  - O(n+k)
  - special case: O(n) when k = O(n)
- Stable sort
  - numbers with the same value appear in the output array in the same order as they do in the input array
- Counting sort is stable
- Counting sort is not in-place

Radix Sort

RADIX_SORT(A,d)

for $i \leftarrow 1$ **to** d

  do use a stable sort to sort array A on digit i

329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Time Complexity (1/2)

Lemma

Given n d-digit numbers in which each digit can take on up to k possible values, RADIX-SORT correctly sorts these number in $\Theta(d(n+k))$ time

When d is a constant and k = O(n), Radix Sort runs in linear time

Radix Sort is not in-place

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Time Complexity (2/2)

Lemma

Given n b-bit numbers and any positive integer $r \le b$, RADIX- SORT correctly sorts these numbers in $\boldsymbol{\Theta((b/r)(n+2^r))}$ time.

Proof : Choose $d = \lceil b/r \rceil$

If $b < \lfloor \lg n \rfloor$ ➔ $2^r < n$ ➔ $(n + 2^r) = \Theta(n)$

choose $r = b$ ➔ $\boldsymbol{\Theta((b/r)(n+2^r)) = \Theta(n)}$

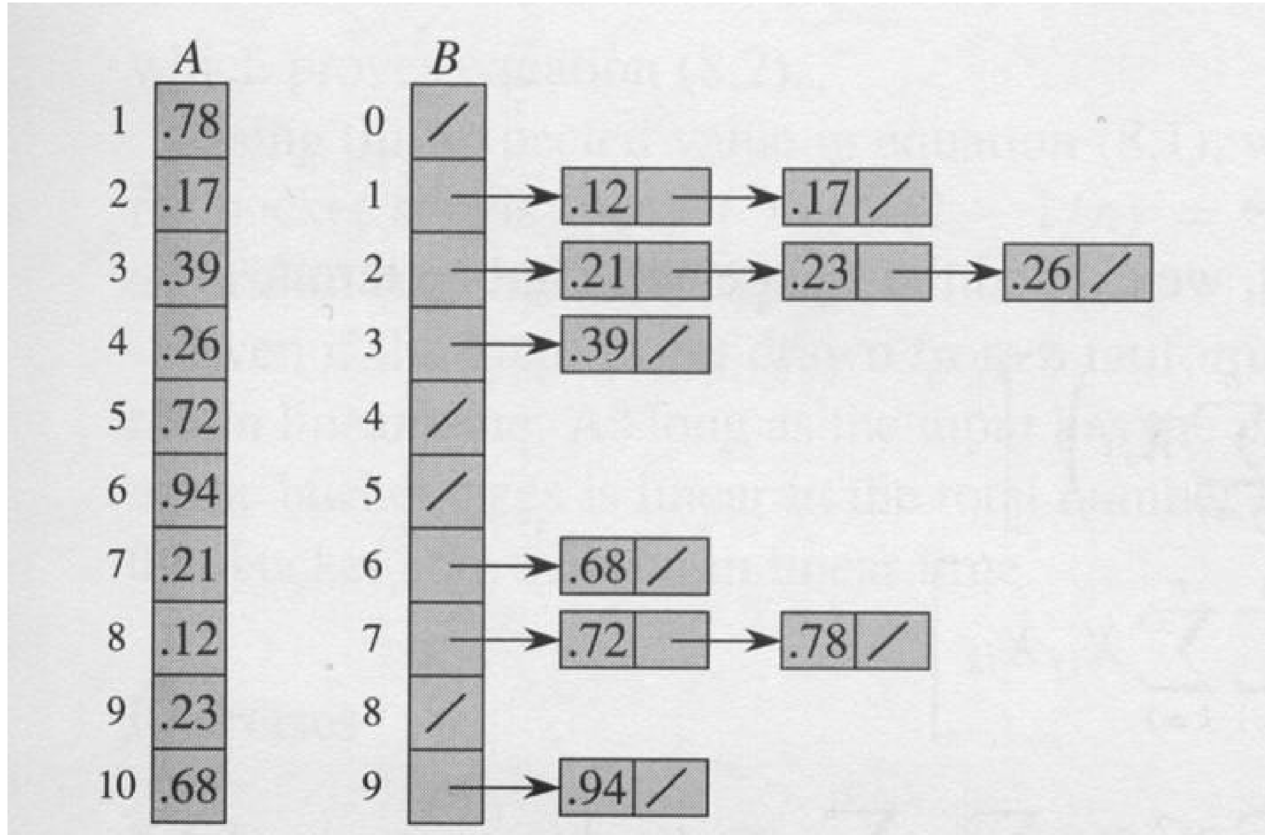
If $b \ge \lfloor \lg n \rfloor$ ➔ choose $r = \lfloor \lg n \rfloor$ ➔ $\Theta((b/r)(n+2^r)) = \Theta(bn/\lg n)$

Sorting in Linear Time

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Bucket Sort

Assumptions
1. 0 ≤ input < 1
2. uniform distribution



BUCKET_SORT(A)

$n \leftarrow length[A]$

for $i \leftarrow 1$ **to** n

  do insert $A[i]$ into list $B[\lfloor nA[i] \rfloor]$

for $i \leftarrow 0$ **to** $n-1$

  do sort list $B[i]$ with **insertion sort**

concatenate $B[0], B[1], \ldots, B[n-1]$ together in order

Time Complexity (1/5)

- Let n_i be the random variable denoting the number of elements placed in bucket B[i]

$$T(n) = \Theta(n) + \boxed{\sum_{i=0}^{n-1} O(n_i^2) \qquad \textbf{due to insertion sort}}$$

$$\begin{aligned} E[T(n)] &= E\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\ &= \Theta(n) + \sum_{i=0}^{n-1} E[O(n_i^2)] \\ &= \Theta(n) + \sum_{i=0}^{n-1} O(E[n_i^2]) \end{aligned}$$

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Sorting in Linear Time

Time Complexity (2/5)

- Define indicator random variables
  X_{ij} = I { A[*j*] falls in bucket *i* },
  for i = 0, 1, …, n-1 and j = 1, 2,…,n

$$\Rightarrow \quad n_i = \sum_{j=1}^{n} X_{ij}$$

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Time Complexity (3/5)

$$E[n_i^2] = E\left[\left(\sum_{j=1}^{n} X_{ij}\right)^2\right]$$

$$= E\left[\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ij} X_{ik}\right]$$

$$= E\left[\sum_{j=1}^{n} X_{ij}^2 + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n\\ k\ne j}} X_{ij} X_{ik}\right]$$

$$= \sum_{j=1}^{n} E[X_{ij}^2] + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n\\ k\ne j}} E[X_{ij} X_{ik}]$$

Sorting in Linear Time

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Time Complexity (4/5)

- Indicator random variable X_{ij}
  - is 1 with probability **1/n**
  - is 0 otherwise

$$\Rightarrow \quad E[X_{ij}^2] = 1 \cdot \frac{1}{n} + 0 \cdot \left(1 - \frac{1}{n}\right) = \frac{1}{n}$$

- When $k \neq j$, the variables X_{ij} and X_{ik} are independent

$$\Rightarrow \quad E[X_{ij} X_{ik}] = E[X_{ij}] E[X_{ik}] = \frac{1}{n} \cdot \frac{1}{n} = \frac{1}{n^2}$$

Time Complexity (5/5)

$$E[n_i^2] = \sum_{j=1}^{n} \frac{1}{n} + \sum_{1\le j\le n} \sum_{\substack{1\le k\le n\\ k\ne j}} \frac{1}{n^2}$$

$$= n \cdot \frac{1}{n} + n(n-1) \cdot \frac{1}{n^2}$$

$$= 1 + \frac{n-1}{n}$$

$$= 2 - \frac{1}{n}$$

$$\Rightarrow E[T(n)] = \Theta(n) + \sum_{i=0}^{n-1} O[E(n_i^2)] = \Theta(n) + n \cdot O(2 - 1/n) = \Theta(n)$$

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw

Summary

Comparison-based sorters				
Algorithm	Runtime			In-place?
	Best case	Average case	Worst case	
Insertion	$O(n)$	$O(n^2)$	$O(n^2)$	Yes
Merge	$O(n \lg n)$	$O(n \lg n)$	$O(n \lg n)$	No
Heap	$O(n \lg n)$	$O(n \lg n)$	$O(n \lg n)$	Yes
Quicksort	$O(n \lg n)$	$O(n \lg n)$	$O(n^2)$	Yes
Non-comparison-based sorters				
Counting	$O(n+k)$	$O(n+k)$	$O(n+k)$	No
Radix	$O(d(n+k'))$	$O(d(n+k'))$	$O(d(n+k'))$	No
Bucket	-	$O(n)$	-	No

Juinn-Dar Huang jdhuang@mail.nctu.edu.tw