

ساختمان داده ها

مرتب سازی شمارشی (Counting Sort)

مدرس: غیاثی شیرازی
دانشگاه فردوسی مشهد

مساله مرتب سازی خاص

- ورودی

– آرایه $A[1..n]$ که هر عنصر آن عددی صحیح بین 0 تا k است.

$(k+1)$ مقدار

- خروجی

– آرایه مرتب $B[1..n]$

الگوریتم مرتب سازی شمارشی

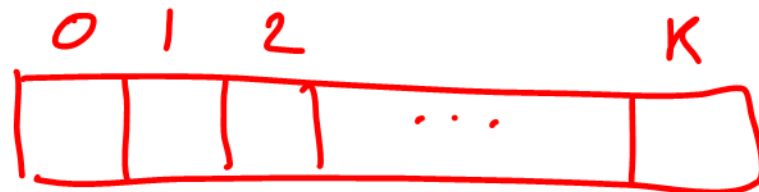
$C[0 \dots k]$

$A[1 \dots n]$
 $B[1 \dots n]$

COUNTING-SORT(A, B, k)

```

1  let  $C[0 \dots k]$  be a new array
2  for  $i = 0$  to  $k$ 
3       $C[i] = 0$ 
4  for  $j = 1$  to  $A.length$ 
5       $C[A[j]] = C[A[j]] + 1$ 
6  //  $C[i]$  now contains the number of elements equal to  $i$ .
7  for  $i = 1$  to  $k$ 
8       $C[i] = C[i] + C[i - 1]$ 
9  //  $C[i]$  now contains the number of elements less than or equal to  $i$ .
10 for  $j = A.length$  downto 1
11      $B[C[A[j]]] = A[j]$ 
12      $C[A[j]] = C[A[j]] - 1$ 
    
```



$C[i]$ کل آفریز عنصر با شماره i به تازگی می دهد.

برای stable بودن الگوریتم
نیاز

نحوه ی اجرا

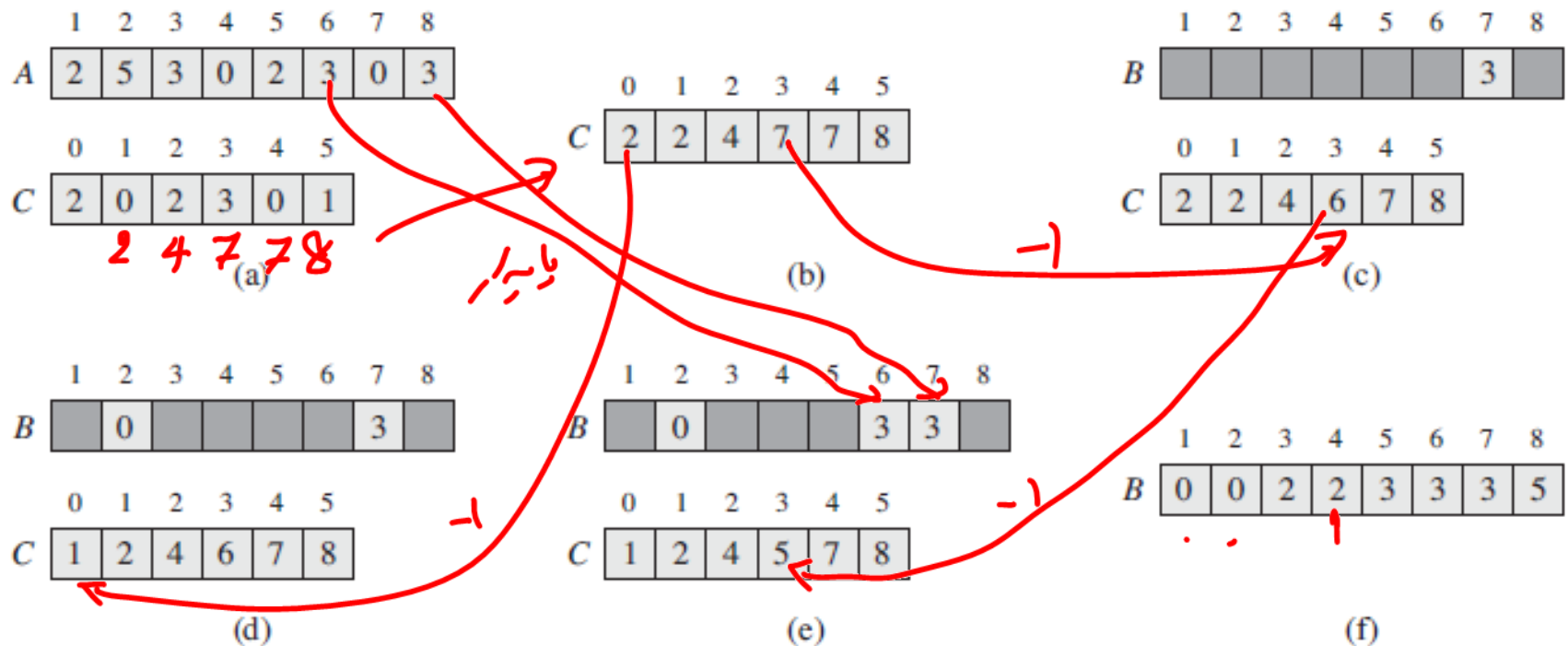
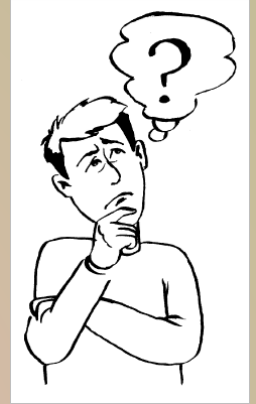


Figure 8.2 The operation of COUNTING SORT on an input array $A[1..8]$, where each element of A is a nonnegative integer no larger than $k = 5$. (a) The array A and the auxiliary array C after line 5. (b) The array C after line 8. (c) (e) The output array B and the auxiliary array C after one, two, and three iterations of the loop in lines 10–12, respectively. Only the lightly shaded elements of array B have been filled in. (f) The final sorted output array B .

تحلیل زمان اجرا



C++ $\Theta(1)$
Java $\Theta(k+1)$

COUNTING-SORT(A, B, k)

```
1 let  $C[0..k]$  be a new array
2 for  $i = 0$  to  $k$  }  $\Theta(k)$ 
3    $C[i] = 0$ 
4 for  $j = 1$  to  $A.length$  }  $\Theta(n)$ 
5    $C[A[j]] = C[A[j]] + 1$ 
6 //  $C[i]$  now contains the number of elements equal to  $i$ .
7 for  $i = 1$  to  $k$  }  $\Theta(k)$ 
8    $C[i] = C[i] + C[i - 1]$ 
9 //  $C[i]$  now contains the number of elements less than or equal to  $i$ .
10 for  $j = A.length$  downto 1 }  $\Theta(n)$ 
11    $B[C[A[j]]] = A[j]$ 
12    $C[A[j]] = C[A[j]] - 1$ 
```

- ✓ زمان اجرای خطوط ۲ و ۳ از مرتبه ؟ می باشد.
- ✓ زمان اجرای خطوط ۴ و ۵ از مرتبه ؟ می باشد.
- ✓ زمان اجرای خطوط ۷ و ۸ از مرتبه ؟ می باشد.
- ✓ زمان اجرای خطوط ۱۰ تا ۱۲ از مرتبه ؟ می باشد.

✓ پس در مجموع زمان اجرای الگوریتم از مرتبه ؟ است.

$\Theta(n+k)$

تحلیل زمان اجرا

COUNTING-SORT(A, B, k)

```
1  let  $C[0..k]$  be a new array
2  for  $i = 0$  to  $k$ 
3       $C[i] = 0$ 
4  for  $j = 1$  to  $A.length$ 
5       $C[A[j]] = C[A[j]] + 1$ 
6  //  $C[i]$  now contains the number of elements equal to  $i$ .
7  for  $i = 1$  to  $k$ 
8       $C[i] = C[i] + C[i - 1]$ 
9  //  $C[i]$  now contains the number of elements less than or equal to  $i$ .
10 for  $j = A.length$  downto 1
11      $B[C[A[j]]] = A[j]$ 
12      $C[A[j]] = C[A[j]] - 1$ 
```

خطوط ۲ و ۳ از مرتبه $\Theta(k)$ می باشد. خطوط ۴ و ۵ از مرتبه $\Theta(n)$ می باشد.

خطوط ۷ و ۸ از مرتبه $\Theta(k)$ می باشد. خطوط ۱۰ تا ۱۲ از مرتبه $\Theta(n)$ می باشد.

پس در مجموع زمان اجرای الگوریتم از مرتبه $\Theta(2n+2k)$ یا همان $\Theta(n+k)$ است.

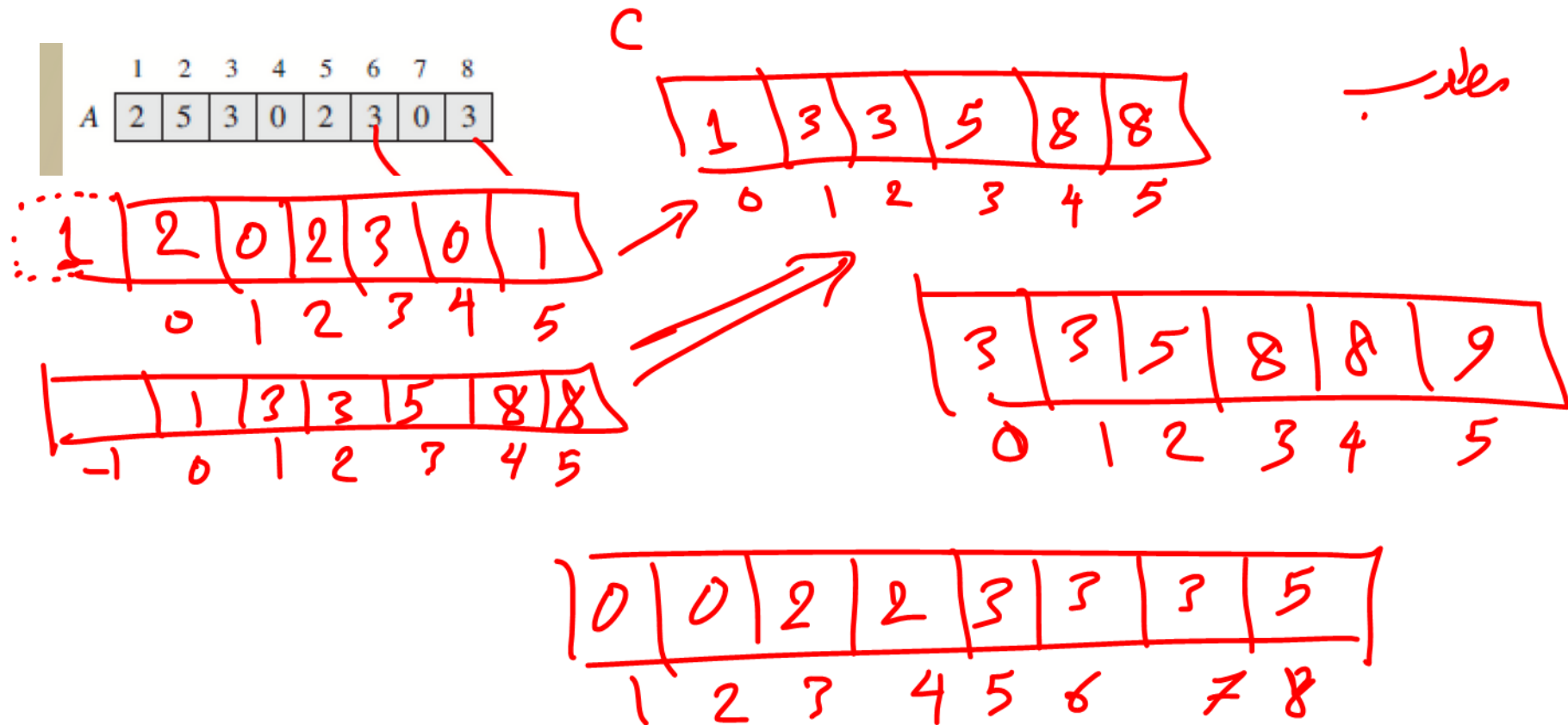
پایداری

■ پایدار است.

■ عناصر مساوی را در آرایه ورودی از انتها به ابتدا می بینیم.

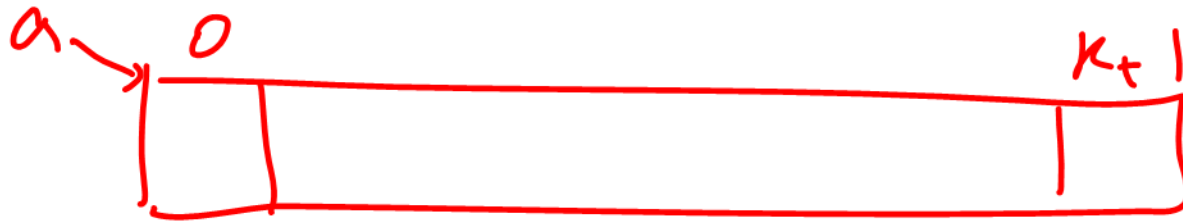
■ در آرایه خروجی نیز عناصر مساوی از انتها به ابتدا قرار داده می شوند.

الگوریتم مرتب‌سازی شمارشی را تغییر دهید؛ به نحوی که داده‌ها را نه از ابتدای آرایه A، بلکه از انتهای آرایه B تراز در هر (بایه ارم بانه)

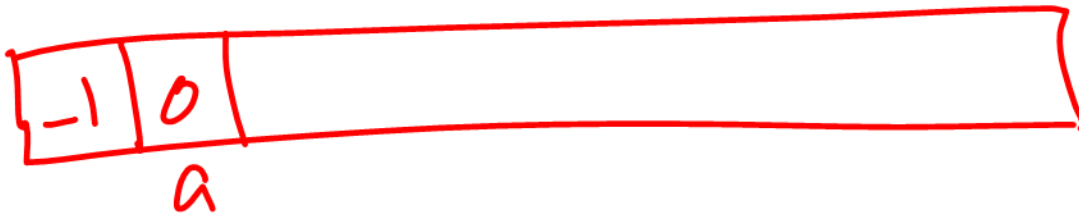


`int* a = new int[k+2]` →

بناظر خانه ۱-



`a++`



`a[-1]`

`a--`

`delete[] a`

	0	1	2	3	4	5
C	2	0	2	3	0	1

0	0	2	2	3	3	3	5
---	---	---	---	---	---	---	---

--	--

[نام خانہ درجہ آکرس . سن
↓
کلیہ

تمرین

- ص ۱۹۷ کتاب CLRS
- سوال 8.2-4

مطالعه بیشتر

- بخش 8.2 کتاب CLRS با شروع از صفحه ۱۹۴

ساختمان داده ها

مرتب سازی مبنایی

مدرس: غیاثی شیرازی
دانشگاه فردوسی مشهد

تاریخچه

- الگوریتم مرتب سازی مبنایی اولین بار در سال ۱۸۹۰ (قبل از اختراع کامپیوتر) برای انجام سرشماری در ایالات متحده آمریکا پیاده سازی شد.
- این الگوریتم در ماشینی که کارش مرتب سازی کارت های سرشماری بود استفاده شد.
- ماشین مرتب کننده مکانیکی می توانست برنامه ریزی شود و کارت ها را بر اساس اطلاعات یک ستون مشخص دسته بندی کند.

قالب یک کارت مورد استفاده در سرشماری ۱۹۰۰ (کارت های سرشماری ۱۸۹۰ برای صرفه جویی فاقد نوشته بودند)

1	2	3	4	W	M	0	1	5	6	Un	0	6	12	0	6	12	Me	NH	VT	OH	MO	IA	SD
5	6	7	8	9	F	10	15	18	1	2	1	7	13	1	7	13	MA	RI	CT	IND	WIS	MD	NH
1	2	3	4	Ch	20	21	25	30	3	MO	2	8	14	2	8	N	NY	NJ	PA	ILL	MIN	ND	KAN
5	6	7	8	Ap	35	40	45	50	3	WI	3	9	15	3	9	F	MD	VA	WVA	OK	TEN	ALA	CLF
1	2	3	4	Ir	55	60	65	70	3	Wd	4	10	16	4	10	DEL	NC	SC	MS	LA	TEX	DRE	WSH
5	6	7	8	75	80	85	90	95+	Un	D	5	11	17+	5	11	DE	GA	FLA	OKL	IT	ARK	IDA	NEV
1	2	3	4	En	OK	0	a	4	17	11	5	Un	15	2	0	US	Un	En	US	Un	En	UTA	ARI
5	6	7	8	Of	NR	1	b	5	01	12	6	NG	20+	3	1	Gr	Ir	Sc	Gr	Ir	Sc	NM	COL
1	2	3	4	2	NW	4	c	6	0	13	7	1	Na	4	Au	Sw	CE	Wa	Sw	CE	Wa	WYO	MNT
5	6	7	8	4	0	7	d	7	1	14	8	2	Pa	5	Sz	Nw	CF	Hu	Nw	CF	Hu	ALK	AB
1	2	3	4	6	12	10	e	8	2	15	9	3	A	6	Po	Dk	Fr	It	Dk	Fr	It	Au	SEA
5	6	7	8	8+	Un	g	f	9	3	16	10	4	Un	0	01	Ru	Bo	01	Ru	Bo	Sz	Po	NS

الگوریتم مرتب سازی مبنایی

- فرض کنیم هر داده عددی d رقمی است.

RADIX-SORT(A, d)

1 **for** $i = 1$ **to** d

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

مثال

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

اثبات درستی با استقرا بر روی تعداد رقم های مرتب شده

- پایه استقرا:

– در گام اول داده ها بر حسب رقم اول مرتب هستند.

- فرض: داده ها بر حسب k رقم سمت راست مرتب هستند.

- حکم: با مرتب سازی بر حسب رقم $k+1$ داده ها بر حسب $k+1$ رقم سمت راست مرتب خواهند بود.

- اثبات: روشن است که داده ها بر حسب رقم $k+1$ مرتب هستند. برای داده هایی که رقم $k+1$ آنها یکسان است، با توجه به پایداری الگوریتم مرتب سازی و فرض استقرا، حکم برقرار است.

تحلیل زمان اجرا

- ورودی الگوریتم مرتب سازی مبنایی n داده d رقمی است.
- هر رقم می تواند k مقدار مختلف داشته باشد
- در صورت استفاده از الگوریتم **مرتب سازی شمارشی** برای مرتب کردن داده ها بر حسب هر رقم، زمان اجرای الگوریتم مرتب سازی مبنایی از مرتبه $\Theta(d(n + k))$ خواهد بود.

زمان اجرا بر روی n عدد صحیح متمایز b بیتی

- n عدد داریم.
- هر عدد b بیت است. ($d=b$)
- هر بیت دو مقدار دارد. ($k=2$)
- چون اعداد متمایز هستند پس $n \leq 2^b$
- طبق تحلیل قبل، زمان اجرای الگوریتم مرتب سازی مبنایی برابر است با:

$$b(n + 2) > n \log n$$

- آیا می توان بهتر عمل کرد؟

ایده: هر r بیت را با هم یک رقم در نظر بگیریم.

- n عدد داریم.
- هر عدد b/r رقم دارد. ($d = b/r$)
- هر رقم 2^r مقدار دارد. ($k = 2^r$)
- طبق تحلیل قبل، زمان اجرای الگوریتم مرتب سازی مبنایی برابر است با:

$$\Theta(d(n + k)) = \Theta\left(\frac{b}{r}(n + 2^r)\right)$$

انتخاب r

- دیدم که زمان اجرای الگوریتم مرتب سازی مبنایی در حالت قبل برابر بود با

$$T(r) = \Theta\left(\frac{b}{r}(n + 2^r)\right)$$

- برای کمینه سازی مقدار فوق نسبت به r مشتق می گیریم و مقدار آن را مسای صفر قرار می دهیم.

$$-\frac{b}{r^2}(n + 2^r) + \frac{b \ln 2}{r} 2^r = 0$$

انتخاب r

$$r \ln 2 2^r = n + 2^r$$

$$\Rightarrow (r \ln 2 - 1)2^r = n$$

$$\Rightarrow \log(r \ln 2 - 1) + r = \log n$$

• با توجه به کوچک بودن مقدار $\log(r \ln 2 - 1)$ نتیجه می گیریم که مقدار بهینه r تقریباً $\log n$ است.

در این صورت زمان اجرای الگوریتم برابر است با:

$$\Theta\left(\frac{b}{r}(n + 2^r)\right) = \Theta\left(\frac{b}{\log n}n\right)$$

تحلیل مرتبه زمانی

- دیدیم که مرتبه زمانی الگوریتم برابر است با

$$\Theta\left(\frac{b}{\log n} n\right)$$

- اگر $b = O(\log n)$ آنگاه زمان اجرای الگوریتم از مرتبه $\Theta(n)$ خواهد بود.

- توجه کنید که شرط $b = O(\log n)$ با $n = \Omega(2^b)$ معادل است و معنای آن این است که تعداد داده ها باید با تعداد کل داده های قابل نمایش با b بیت متناسب باشد.

تمرین

- ص ۱۹۹ کتاب CLRS
- تمرین 8.3-1
- تمرین 8.3-4

مطالعه بیشتر

- بخش 8.3 با شروع از صفحه ۱۹۷ کتاب CLRS

ساختمان داده ها

مرتب سازی سطلی (Bucket Sort)

مدرس: غیاثی شیرازی
دانشگاه فردوسی مشهد

مساله مرتب سازی خاص

- ورودی

– آرایه $A[1..n]$ که هر عنصر آن مقداری تصادفی با توزیع یکنواخت در بازه $[0,1)$ است.

- خروجی

– آرایه مرتب $B[1..n]$

الگوریتم مرتب سازی سطلی

BUCKET-SORT(A)

1 let $B[0..n-1]$ be a new array

2 $n = A.length$

3 for $i = 0$ to $n-1$

4 make $B[i]$ an empty list

5 for $i = 1$ to n

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

7 for $i = 0$ to $n-1$

8 sort list $B[i]$ with insertion sort

9 concatenate the lists $B[0], B[1], \dots, B[n-1]$ together in order

$$B = \begin{matrix} & h & 0 & 1 & & n-1 \\ \left[\begin{matrix} \\ \left[\right] \\ \left[\hat{0}, \frac{1}{n} \right) \end{matrix} \right] & \left[\begin{matrix} \\ \left[\right] \\ \left[\frac{1}{n}, \frac{2}{n} \right) \end{matrix} \right] & \left[\begin{matrix} \\ \left[\right] \\ \left[\frac{2}{n}, \frac{3}{n} \right) \end{matrix} \right] & \dots & \left[\begin{matrix} \\ \left[\right] \\ \left[\frac{n-1}{n}, 1 \right) \end{matrix} \right] \end{matrix}$$

$$0 \leq A[i] < 1$$
$$0 \leq nA[i] < n$$

$$\lfloor nA[i] \rfloor$$

$$\{0, 1, \dots, n-1\}$$

نحوه ی اجرا

$n = 10$

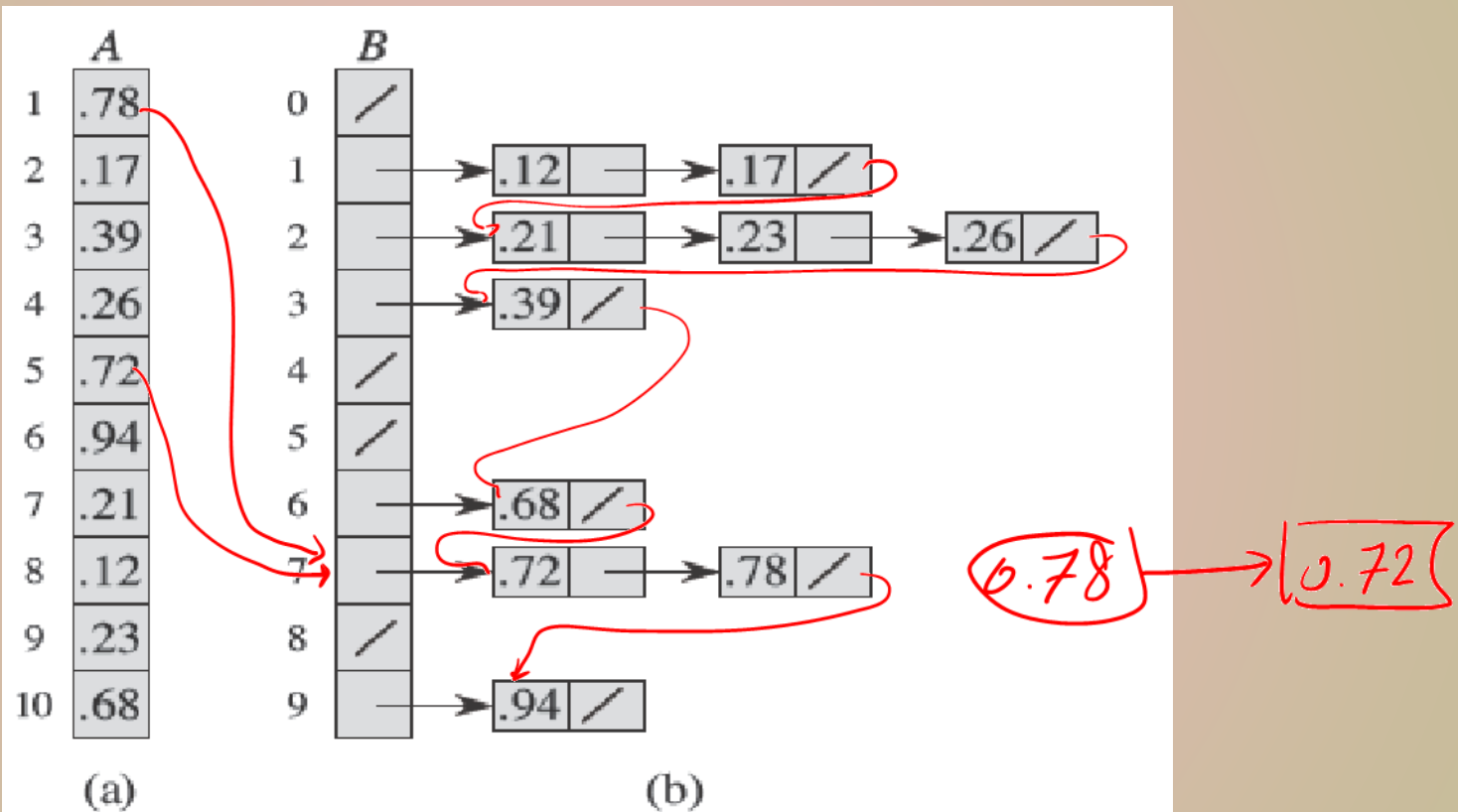


Figure 8.4 The operation of BUCKET SORT for $n = 10$. (a) The input array $A[1 \dots 10]$. (b) The array $B[0 \dots 9]$ of sorted lists (buckets) after line 8 of the algorithm. Bucket i holds values in the half open interval $[i/10, (i + 1)/10)$. The sorted output consists of a concatenation in order of the lists $B[0], B[1], \dots, B[9]$.

پایداری

- با توجه به استفاده از مرتب سازی درجی در گام درونی، این الگوریتم پایدار است.

زمان اجرا

- در بدترین حالت زمان اجرا از مرتبه $\Theta(n^2)$ است.
- زمان اجرای متوسط از مرتبه $\Theta(n)$ است.

تحلیل زمان اجرای متوسط

اگر $A[n]$ سرخانه نامیده

$$X_{ij} = \begin{cases} 1 \\ 0 \end{cases}$$

در غیر این صورت

$$T = \sum_{i=1}^n \left(\sum_{j=1}^n X_{ij} \right)^2$$

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

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

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

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

$$E[X_{ij}] = \frac{1}{n}$$

$$= \sum_{i=1}^n \left[\sum_{j=1}^n E[\underbrace{X_{ij}^2}_{X_{ij}}] + \sum_{j=1}^n \sum_{\substack{k=1 \\ k \neq j}}^n E[X_{ij}] E[X_{ik}] \right]$$

$$= \sum_{i=1}^n \left[\underbrace{\sum_{j=1}^n \frac{1}{n}}_1 + \underbrace{\sum_{j=1}^n \sum_{\substack{k=1 \\ k \neq j}}^n \frac{1}{n^2}}_{\frac{n(n-1)}{n^2}} \right] = \sum_{i=1}^n \left(2 - \frac{1}{n} \right) = 2n - 1 \in \mathcal{O}(n)$$

تمرین

- ص ۲۰۴ کتاب CLRS
- سوال 8.4-1
- سوال 8.4-2
- سوال 8.4-4

مطالعه بیشتر

- بخش 8.4 کتاب CLRS با شروع از صفحه ۲۰۰