بسم الله الرحمن الرحیم

ساختمان‌های داده

جلسه ۲۸

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

○ مسئله مرتب‌سازی:

Input: A sequence of n numbers $\langle a_1, a_2, \ldots, a_n \rangle$.

Output: A permutation (reordering) $\langle a'_1, a'_2, \ldots, a'_n \rangle$ of the input sequence such that $a'_1 \leq a'_2 \leq \cdots \leq a'_n$.

مسئله مرتب‌سازی

○ مسئله مرتب‌سازی:

Input: A sequence of n numbers $\langle a_1, a_2, \ldots, a_n \rangle$.

Output: A permutation (reordering) $\langle a'_1, a'_2, \ldots, a'_n \rangle$ of the input sequence such that $a'_1 \leq a'_2 \leq \cdots \leq a'_n$.

○ مرتب‌سازی مقایسه‌ای:

A comparison sort uses only comparisons between elements to gain order information about an input sequence $\langle a_1, a_2, \ldots, a_n \rangle$. That is, given two elements a_i and a_j, it performs one of the tests $a_i < a_j$, $a_i \leq a_j$, $a_i = a_j$, $a_i \geq a_j$, or $a_i > a_j$ to determine their relative order.

مسئله مرتب‌سازی

○ مرتب‌سازی غیرمقایسه‌ای:

- بدون مقایسه کلیدهای عناصر باهم، مرتب‌سازی می‌کند.
- مانند شمارشی (counting)، مبنایی (radix) و سطلی (bucket)

○ پیچیدگی این الگوریتم‌ها؟

Counting sort

Counting sort assumes that each of the n input elements is an integer in the range 0 to k, for some integer k.

Counting sort

Counting-Sort(A, n, k)

1   let $B[1:n]$ and $C[0:k]$ be new arrays
2   **for** $i = 0$ **to** k
3       $C[i] = 0$
4   **for** $j = 1$ **to** n
5       $C[A[j]] = C[A[j]] + 1$
6   // $C[i]$ now contains the number of elements equal to i.

	1	2	3	4	5	6	7	8
A	2	5	3	0	2	3	0	3

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

Counting sort

COUNTING-SORT(A, n, k)

1   let $B[1:n]$ and $C[0:k]$ be new arrays
2   **for** $i = 0$ **to** k
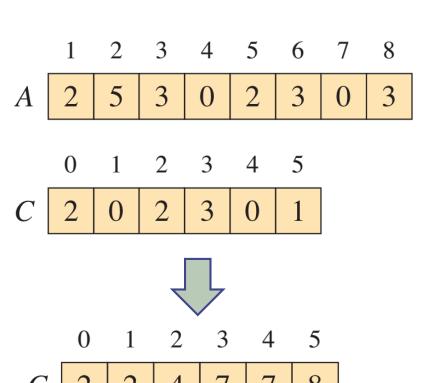3 $C[i] = 0$
4   **for** $j = 1$ **to** n
5      $C[A[j]] = C[A[j]] + 1$
6   // $C[i]$ now contains the number of elements equal to i.
7   **for** $i = 1$ **to** k
8      $C[i] = C[i] + C[i-1]$
9   // $C[i]$ now contains the number of elements less than or equal to i.

	1	2	3	4	5	6	7	8
A	2	5	3	0	2	3	0	3

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

	0	1	2	3	4	5
C	2	2	4	7	7	8

Mojtaba Khalili

Counting sort

A

1	2	3	4	5	6	7	8
2	5	3	0	2	3	0	3

COUNTING-SORT(A, n, k)

1 let $B[1:n]$ and $C[0:k]$ be new arrays
2 **for** $i = 0$ **to** k
3      $C[i] = 0$
4 **for** $j = 1$ **to** 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
8      $C[i] = C[i] + C[i-1]$
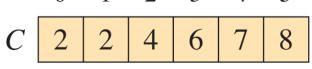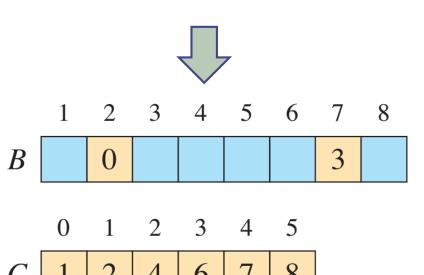9 // $C[i]$ now contains the number of elements less than or equal to i.

B

1	2	3	4	5	6	7	8
						3	

C

0	1	2	3	4	5
2	2	4	6	7	8

B

1	2	3	4	5	6	7	8
	0					3	

C

0	1	2	3	4	5
1	2	4	6	7	8

Counting sort

$$A \quad \begin{array}{|c|c|c|c|c|c|c|c|} \hline 2 & 5 & 3 & 0 & 2 & 3 & 0 & 3 \\ \hline \end{array}$$

positions: 1 2 3 4 5 6 7 8

COUNTING-SORT(A, n, k)

1 let $B[1:n]$ and $C[0:k]$ be new arrays
2 **for** $i = 0$ **to** k
3     $C[i] = 0$
4 **for** $j = 1$ **to** 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
8     $C[i] = C[i] + C[i-1]$
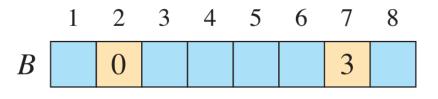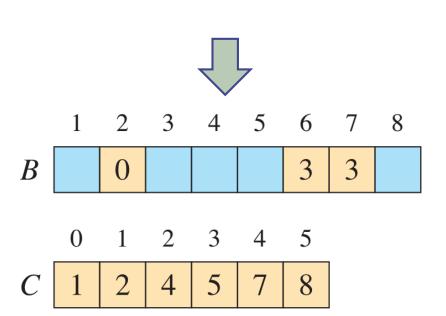9 // $C[i]$ now contains the number of elements less than or equal to i.

B (indices 1–8): position 2 = 0, position 7 = 3

C (indices 0–5):
$$\begin{array}{|c|c|c|c|c|c|} \hline 1 & 2 & 4 & 6 & 7 & 8 \\ \hline \end{array}$$

B (indices 1–8): position 2 = 0, position 6 = 3, position 7 = 3

C (indices 0–5):
$$\begin{array}{|c|c|c|c|c|c|} \hline 1 & 2 & 4 & 5 & 7 & 8 \\ \hline \end{array}$$

Counting sort

$$
\begin{array}{c|cccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
A & 2 & 5 & 3 & 0 & 2 & 3 & 0 & 3 \\
\end{array}
$$

COUNTING-SORT(A, n, k)

1 let $B[1:n]$ and $C[0:k]$ be new arrays
2 **for** $i = 0$ **to** k
3      $C[i] = 0$
4 **for** $j = 1$ **to** 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
8      $C[i] = C[i] + C[i-1]$
9 // $C[i]$ now contains the number of elements less than or equal to i.
10 // Copy A to B, starting from the end of A.
11 **for** $j = n$ **downto** 1
12      $B[C[A[j]]] = A[j]$
13      $C[A[j]] = C[A[j]] - 1$ // to handle duplicate values
14 **return** B

$$
\begin{array}{c|cccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
B & 0 & 0 & 2 & 2 & 3 & 3 & 3 & 5 \\
\end{array}
$$

Counting sort

Counting sort assumes that each of the n input elements is an integer in the range 0 to k, for some integer k. It runs in $\Theta(n + k)$ time, so that when $k = O(n)$, counting sort runs in $\Theta(n)$ time.