



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

ECSE207L

DATA STRUCTURES

Dr. Tapas Badal
Dept. of CSE
Bennett University

The Sorting Problem



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

Input:

A sequence of n numbers a_1, a_2, \dots, a_n

Output:

A permutation (reordering) a_1', a_2', \dots, a_n' of the input

sequence such that $a_1' \leq a_2' \leq \dots \leq a_n'$

Insertion Sort



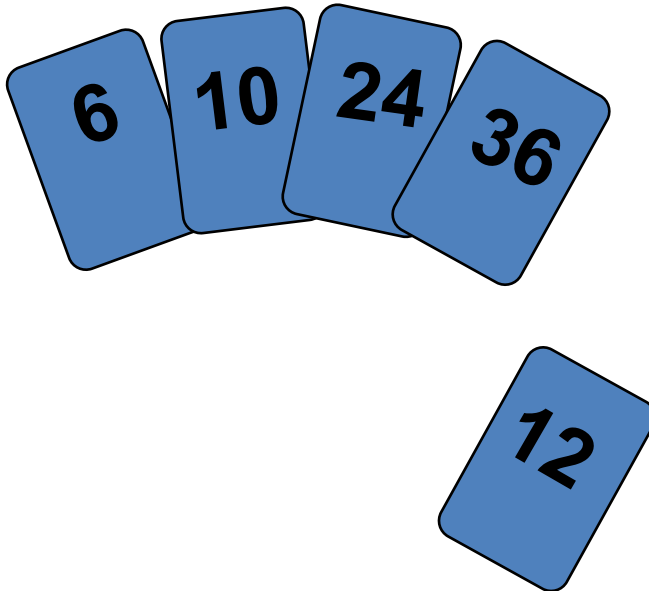
BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

- Idea: like sorting a hand of playing cards
 - Start with an empty left hand and the cards facing down on the table.
 - Remove one card at a time from the table, and insert it into the correct position in the left hand
 - compare it with each of the cards already in the hand, from right to left
 - The cards held in the left hand are sorted
 - these cards were originally the top cards of the pile on the table

Insertion Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

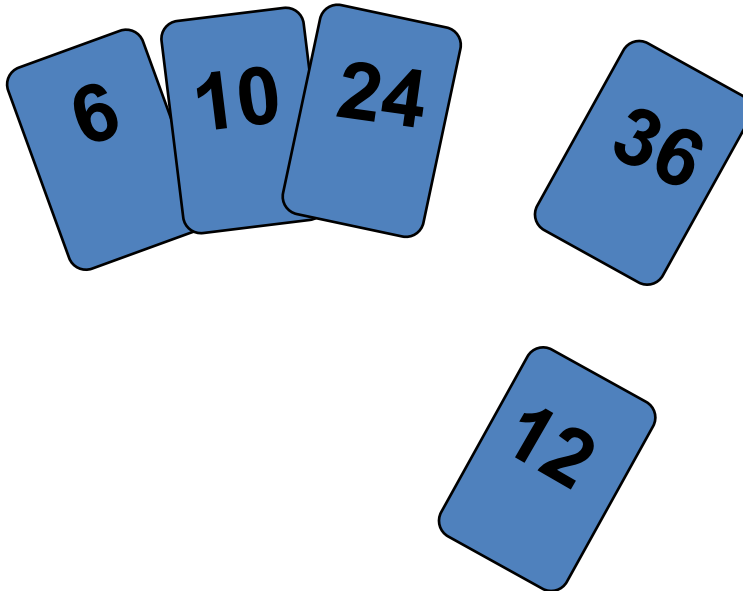


To insert 12, we need to make room for it by moving first 36 and then 24.

Insertion Sort



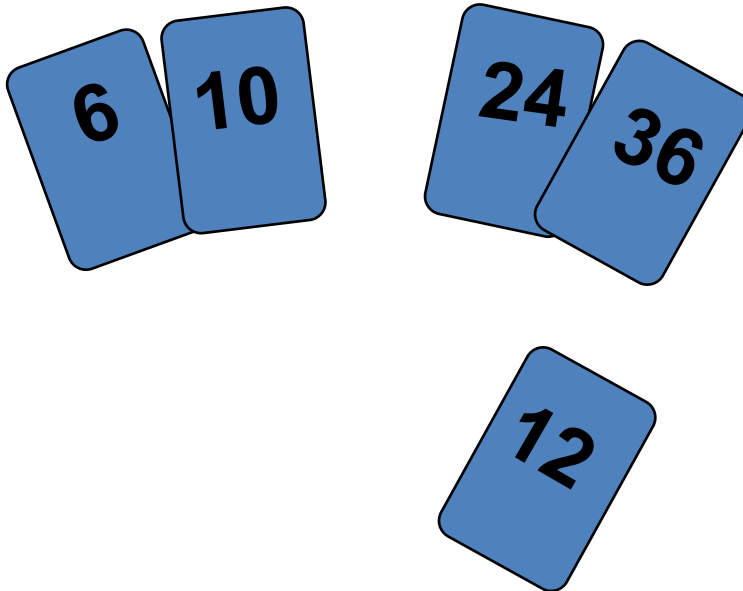
BENNETT
UNIVERSITY
TIMES OF INDIA GROUP



Insertion Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP



Insertion Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

input array

5 2 4 6 1 3

at each iteration, the array is divided in two sub-arrays:

left sub-array

right sub-array

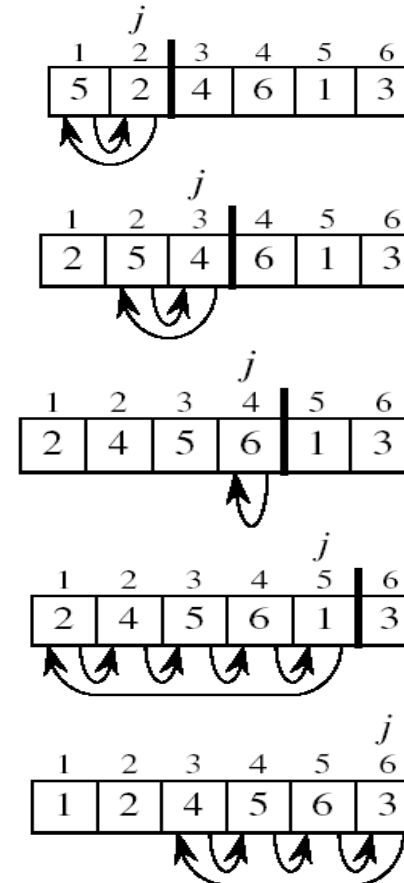
sorted

unsorted

Insertion Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP



Insertion Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

Alg.: INSERTION-SORT(A)

for $j \leftarrow 2$ to n

do $key \leftarrow A[j]$

Insert $A[j]$ into the sorted sequence $A[1 \dots j-1]$

$i \leftarrow j - 1$

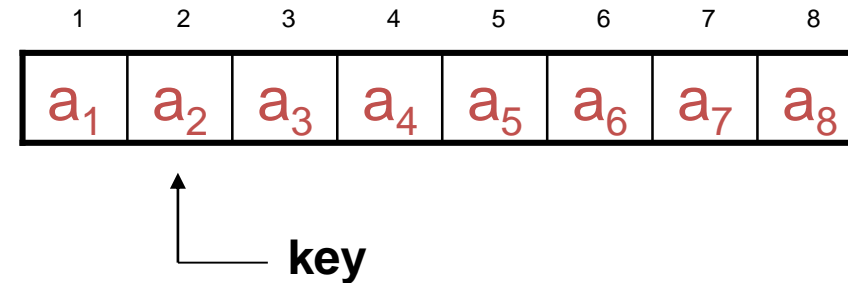
while $i > 0$ and $A[i] > key$

do $A[i + 1] \leftarrow A[i]$

$i \leftarrow i - 1$

$A[i + 1] \leftarrow key$

- Insertion sort – sorts the elements in place

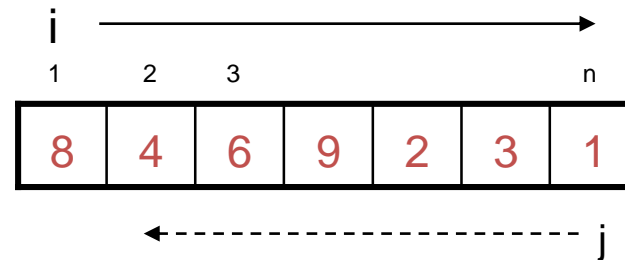


Bubble Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP





- Idea:
 - Repeatedly pass through the array
 - Swaps adjacent elements that are out of order



- Easier to implement, but slower than Insertion sort



Bubble sort example

Initial	<table><tr><td>5</td><td>3</td><td>8</td><td>4</td><td>6</td></tr></table>	5	3	8	4	6	Initial Unsorted array
5	3	8	4	6			
Step 1	<table><tr><td>5</td><td>3</td><td>8</td><td>4</td><td>6</td></tr></table> 	5	3	8	4	6	Compare 1 st and 2 nd (Swap)
5	3	8	4	6			
Step 2	<table><tr><td>3</td><td>5</td><td>8</td><td>4</td><td>6</td></tr></table> 	3	5	8	4	6	Compare 2 nd and 3 rd (Do not Swap)
3	5	8	4	6			
Step 3	<table><tr><td>3</td><td>5</td><td>8</td><td>4</td><td>6</td></tr></table> 	3	5	8	4	6	Compare 3 rd and 4 th (Swap)
3	5	8	4	6			
Step 4	<table><tr><td>3</td><td>5</td><td>4</td><td>8</td><td>6</td></tr></table> 	3	5	4	8	6	Compare 4 th and 5 th (Swap)
3	5	4	8	6			
Step 5	<table><tr><td>3</td><td>5</td><td>4</td><td>6</td><td>8</td></tr></table>	3	5	4	6	8	Repeat Step 1-5 until no more swaps required
3	5	4	6	8			

Bubble Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

Alg.: BUBBLESORT(A)
 for $i \leftarrow 1$ to $\text{length}[A]-1$
 do for $j \leftarrow 1$ to $\text{length}[A] - i$
 do if $A[j] > A[j+1]$
 then exchange $A[j] \leftrightarrow A[j+1]$

Selection Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

- Idea:
 - Find the smallest element in the array
 - Exchange it with the element in the first position
 - Find the second smallest element and exchange it with the element in the second position
 - Continue until the array is sorted
- Disadvantage:
 - Running time depends only slightly on the amount of order in the file

Example



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

8	4	6	9	2	3	1
---	---	---	---	---	---	---

1	4	6	9	2	3	8
---	---	---	---	---	---	---

1	2	6	9	4	3	8
---	---	---	---	---	---	---

1	2	3	9	4	6	8
---	---	---	---	---	---	---

1	2	3	4	9	6	8
---	---	---	---	---	---	---

1	2	3	4	6	9	8
---	---	---	---	---	---	---

1	2	3	4	6	8	9
---	---	---	---	---	---	---

1	2	3	4	6	8	9
---	---	---	---	---	---	---

Selection Sort



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

Alg.: SELECTION-SORT(A)

$n \leftarrow \text{length}[A]$

for $j \leftarrow 1$ to $n - 1$

 do $\text{smallest} \leftarrow j$

 for $i \leftarrow j + 1$ to n

 do if $A[i] < A[\text{smallest}]$

 then $\text{smallest} \leftarrow i$

 exchange $A[j] \leftrightarrow A[\text{smallest}]$



Sorting Algorithm	Worst Case	Best Case	Average Case
Bubble Sort	$O(n^2)$	$O(n)$	$O(n^2)$
Insertion Sort	$O(n^2)$	$O(n)$	$O(n^2)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$

Exercise:



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

Suppose you have the following list of numbers to sort: [19, 1, 9, 7, 3, 10, 13, 15, 8, 12] which list represents the partially sorted list after three complete passes of bubble sort?

- (A) [1, 9, 19, 7, 3, 10, 13, 15, 8, 12]
- (B) [1, 3, 7, 9, 10, 8, 12, 13, 15, 19]
- (C) [1, 7, 3, 9, 10, 13, 8, 12, 15, 19]
- (D) [1, 9, 19, 7, 3, 10, 13, 15, 8, 12]

Alg.: BUBBLESORT(A)

for $i \leftarrow 1$ to $\text{length}[A]$

do for $j \leftarrow \text{length}[A]$ downto $i + 1$

do if $A[j] < A[j - 1]$

then exchange $A[j] \leftrightarrow A[j - 1]$



BENNETT
UNIVERSITY
TIMES OF INDIA GROUP

THANKYOU

@csebennett



cse_bennett

