Insertion Sort Algorithms

CS 601 – Advanced Algorithms

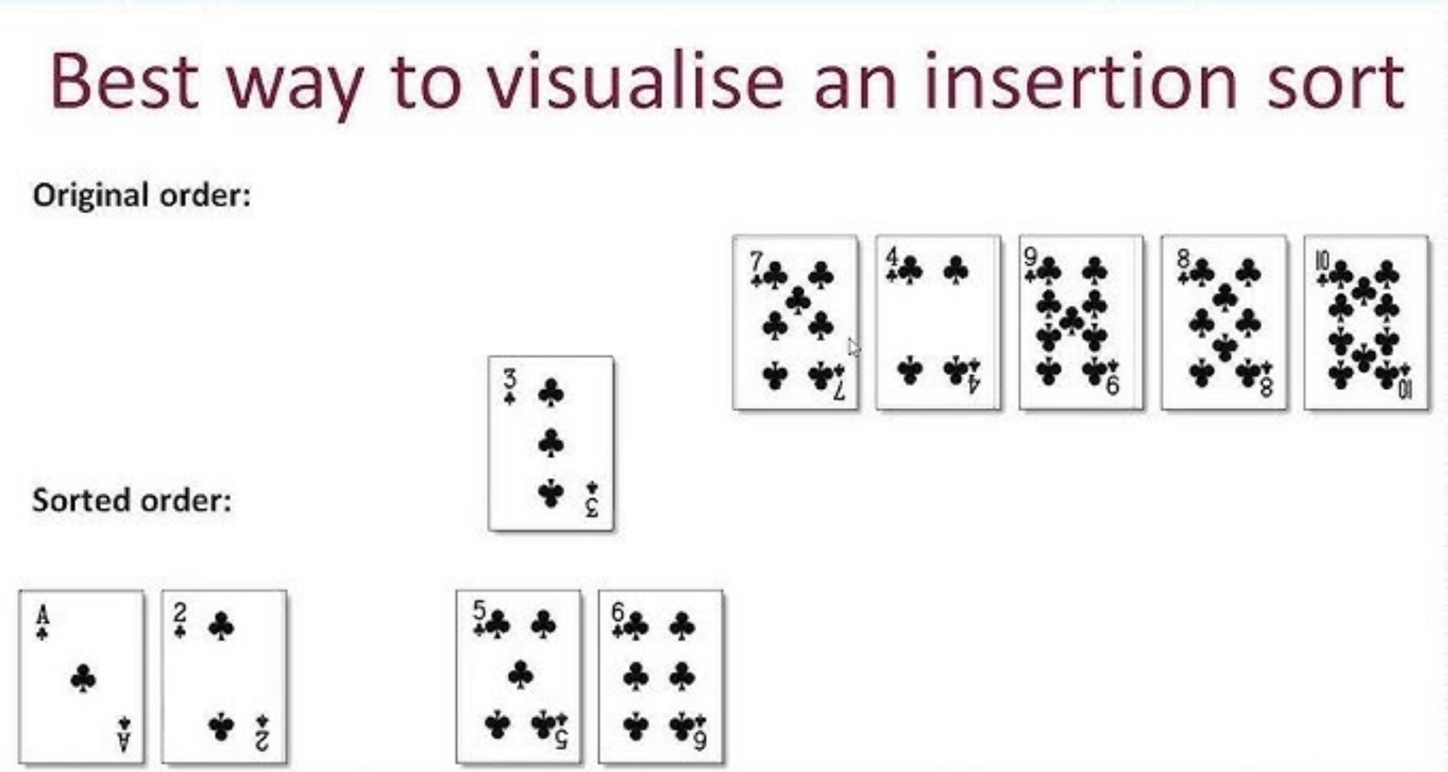
Presented by Jolene Le

Instructor: Professor Zahra Derakhshandeh

Wednesday, August 17th, 2024

Introduction

- Insertion sort is a simple sorting algorithm that works by iteratively inserting each element of an unsorted list into its correct position in a sorted portion of the list.
- It is considered an “in-place” sorting algorithm.
- It is a stable sorting algorithm.



Sorting problems

- Input: A sequence of n numbers $[a_1, a_2, \ldots, a_n]$
- Output: A permutation (reordering) $[a'_1, a'_2, \ldots, a'_n]$ of the input sequence such that $a'_1 \leq a'_2 \leq a'_n$
- Keys: numbers to be sorted
- Satellite data: data associated with keys
- Record: formed by a key and satellite data
- Example: Student records in a spreadsheet

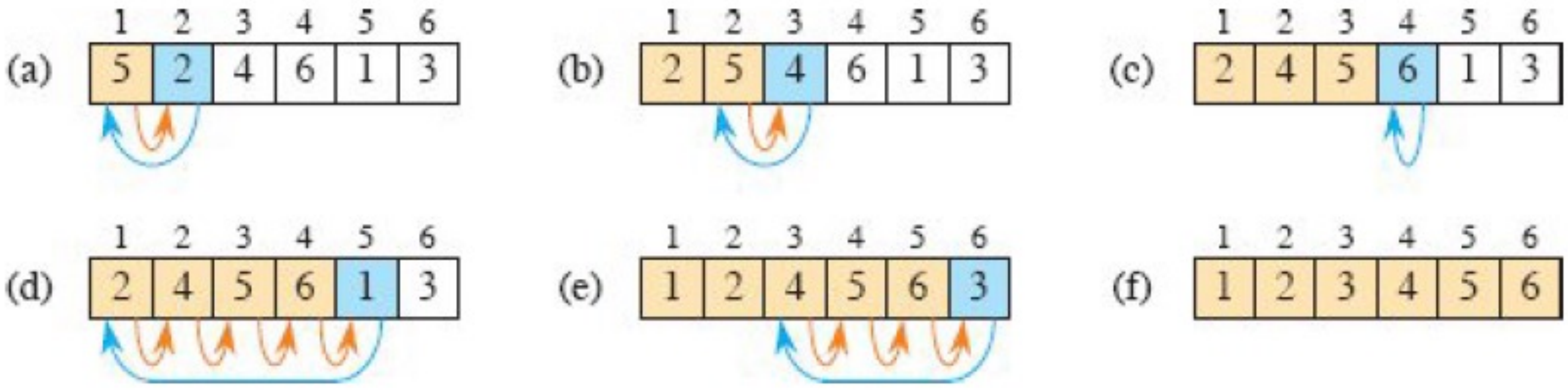
Algorithms

- First element in the array is sorted
- Start with the second element.
- Compare with the sorted ones (to its left).
- Shift all larger elements one position to the right.
- Insert to its correct position.
- Repeat until the entire array is sorted.

```
INSERTION_SORT(A, n)
    for i = 2 to n
        key = A[i]
// Insert A[i] into the sorted subarray A[1 : i - 1].
        j = i - 1
        while j > 0 and A[j] > key
            A[j + 1] = A[j]
            j = j - 1
        A[j + 1] = key

```

Instance [5, 2, 4, 6, 1, 3]




```
INSERTION-SORT(A, n)
 for i = 2 to n
    key = A[i]
    j = i - 1
    while j > 0 and A[j] > key
        A[j + 1] = A[j]
        j = j - 1
    A[j + 1] = key
```

Instance [36, 42, 51, 27, 42, 49]

36	42	51	27	42	49	
36	42	51	27	42	49	i = 2, key = 42
36	42	51	27	42	49	i = 3, key = 51
	36	42	51	42	49	i = 4, key = 27
27	36	42	51	42	49	
27	36	42		51	49	i = 5, key = 42
27	36	42	42	51	49	
27	36	42	42		51	i = 6, key = 51
27	36	42	42	49	51	

Insertion sort Descending - [36, 42, 51, 27, 42, 49]

```
INSERTION_SORT_DECREASING(A, n)
 for i = 2 to n
    key = A[i]
    j = i - 1
    while j>0 and A[j] < key
        A[j + 1] = A[j]
        j = j - 1
    A[j + 1] = key
```

36	42	51	27	42	49	
42	36	51	27	42	49	i = 2, key = 42
51	42	36	27	42	49	i = 3, key = 51
51	42	36	27	42	49	i = 4, key = 27
51	42		36	27	49	i = 5, key = 42
51	42	42	36	27	49	
51		42	42	36	27	i = 6, key = 51
51	49	42	42	36	27	

Analyzing Algorithms – Correctness

- Initialization: i=2, subarray **A[1 : i – 1]** consists of A[1] → first property of iteration holds
- Maintenance: A[i – 1], A[i – 2], A[i – 3], …, A[i], subarray consists of **A[1:i]** in sorted order → second property of the outer loop holds
- Termination: i = n + 1, subarray **A[1:n]** consists of A[1:n] in sorted order → algorithm is correct

```
INSERTION-SORT(A, n)
 for i = 2 to n
    key = A[i]
    j = i - 1
    while j > 0 and A[j] > key
        A[j + 1] = A[j]
        j = j - 1
    A[j + 1] = key
```

Analyzing Algorithms – Time complexity

for $i = 2$ **to** n	c_1	n
$key = A[i]$	c_2	$n-1$
// Insert $A[i]$ into the sorted subarray $A[1:i-1]$.	0	$n-1$
$j = i-1$	c_4	$n-1$
while $j > 0$ and $A[j] > key$	c_5	$\sum_{i=2}^{n} t_i$
$A[j+1] = A[j]$	c_6	$\sum_{i=2}^{n} (t_i - 1)$
$j = j-1$	c_7	$\sum_{i=2}^{n} (t_i - 1)$
$A[j+1] = key$	c_8	$n-1$

$$
\begin{aligned}
T(n) = {} & c_1 n + c_2(n-1) + c_4(n-1) + c_5 \sum_{i=2}^{n} t_i + c_6 \sum_{i=2}^{n} (t_i - 1) \\
& + c_7 \sum_{i=2}^{n} (t_i - 1) + c_8(n-1) .
\end{aligned}
$$

Analyzing Algorithms – Time complexity

for $i = 2$ **to** n	c_1	n
$key = A[i]$	c_2	$n-1$
// Insert $A[i]$ into the sorted subarray $A[1:i-1]$.	0	$n-1$
$j = i - 1$	c_4	$n-1$
while $j > 0$ and $A[j] > key$	c_5	$\sum_{i=2}^{n} t_i$
$A[j+1] = A[j]$	c_6	$\sum_{i=2}^{n} (t_i - 1)$
$j = j - 1$	c_7	$\sum_{i=2}^{n} (t_i - 1)$
$A[j+1] = key$	c_8	$n-1$

$$
\begin{aligned}
T(n) &= c_1 n + c_2(n-1) + c_4(n-1) + c_5 \sum_{i=2}^{n} t_i + c_6 \sum_{i=2}^{n} (t_i - 1) \\
&\quad + c_7 \sum_{i=2}^{n} (t_i - 1) + c_8(n-1) .
\end{aligned}
$$

$$
\begin{aligned}
T(n) &= c_1 n + c_2(n-1) + c_4(n-1) + c_5(n-1) + c_8(n-1) \\
&= (c_1 + c_2 + c_4 + c_5 + c_8)n - (c_2 + c_4 + c_5 + c_8) .
\end{aligned}
$$

Best case

Analyzing Algorithms – Time complexity – Worst case

Given:

$$\sum_{i=2}^{n} i = \left(\sum_{i=1}^{n} i\right) - 1 = \frac{n(n+1)}{2} - 1$$

and

$$\sum_{i=2}^{n} (i-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

for $i = 2$ **to** n	c_1	n
$key = A[i]$	c_2	$n-1$
// Insert $A[i]$ into the sorted subarray $A[1:i-1]$.	0	$n-1$
$j = i - 1$	c_4	$n-1$
while $j > 0$ and $A[j] > key$	c_5	$\sum_{i=2}^{n} t_i$
$A[j+1] = A[j]$	c_6	$\sum_{i=2}^{n} (t_i - 1)$
$j = j - 1$	c_7	$\sum_{i=2}^{n} (t_i - 1)$
$A[j+1] = key$	c_8	$n-1$

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5 \sum_{i=2}^{n} t_i + c_6 \sum_{i=2}^{n} (t_i - 1) + c_7 \sum_{i=2}^{n} (t_i - 1) + c_8(n-1).$$

$$\begin{aligned} T(n) &= c_1 n + c_2(n-1) + c_4(n-1) + c_5\left(\frac{n(n+1)}{2} - 1\right) \\ &\quad + c_6\left(\frac{n(n-1)}{2}\right) + c_7\left(\frac{n(n-1)}{2}\right) + c_8(n-1) \\ &= \left(\frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2}\right)n^2 + \left(c_1 + c_2 + c_4 + \frac{c_5}{2} - \frac{c_6}{2} - \frac{c_7}{2} + c_8\right)n \\ &\quad - (c_2 + c_4 + c_5 + c_8). \end{aligned}$$

$$\Rightarrow\ T(n) = an^2 + b + c$$

Analyzing Algorithms – Time complexity Summary

```
INSERTION-SORT(A, n)
 for i = 2 to n
    key = A[i]
    j = i - 1
    while j > 0 and A[j] > key
        A[j + 1] = A[j]
        j = j - 1
    A[j + 1] = key
```

O(n)

O(i)

$$T(n) = \sum_{i=1}^{n} O(i) = O(2 + 3 + 4 + \ \ldots + n)$$

$$T(n) = \ O\left(\frac{n(n-1)}{2}\right)$$

$$T(n) = \ O\left(\frac{1}{2}n^2 + \frac{1}{2}n\right) = O\ (n^2)$$

- Best case: $\mathrm{O(n)}$
- Worst case: $\mathrm{O(n^2)}$
- Average case: $\mathrm{O(n^2)}$

Reference

- T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms* (4th Edition), TheMIT Press, 2022