

# Bubble Sort

(A Taste of Algorithms: Definition, Design, and Analysis)

Hengfeng Wei

Institute of Computer Software  
Nanjing University

December 18, 2016



# Bubble Sort

- 1 Sorting
- 2 Bubble Sort
- 3 Analysis

# Bubble Sort

- 1 Sorting
- 2 Bubble Sort
- 3 Analysis

# Sorting

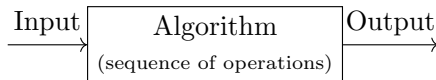
The sorting problem:

**Input:** A sequence of  $n$  integers  $A$ :  $a_1 a_2 \cdots a_n$ .

**Output:**  $A$  sorted (non-decreasing order).

$$3 \ 1 \ 4 \ 2 \implies 1 \ 2 \ 3 \ 4$$

# Algorithms



## Correctness!

**Definiteness:** precisely defined operations

**Finiteness:** termination

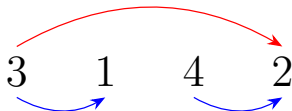
**Effectiveness:** a reasonable model; basic operations

- for sorting: compare, swap

# Inversions

$$A = a_1 \quad a_2 \quad \dots \quad a_n.$$

If  $i < j$  and  $a_i > a_j$ , then  $(a_i, a_j)$  is an **inversion**.



**Adjacent** inversion:  $j = i + 1$

# Inversions

$A$  is sorted  $\implies A$  has no inversions  
 $\implies A$  has no adjacent inversions.

$A$  has no adjacent inversions  $\implies \forall i \in [1, n-1] : a_i \leq a_{i+1}$   
 $\implies A$  is sorted.

$A$ is sorted $\iff A$ has no adjacent inversions.
--

# Bubble Sort

- 1 Sorting
- 2 Bubble Sort
- 3 Analysis



# Bubble Sort

Basic idea: to eliminate all adjacent inversions

---

---

```
1: repeat
2:   pick any  $i$                                 ▷ Definiteness!
3:   if  $a_i > a_{i+1}$  then
4:     SWAP( $a_i, a_{i+1}$ )
5: until no adjacent inversions                ▷ Finiteness! Definiteness!
```

---

# Bubble Sort: Definiteness

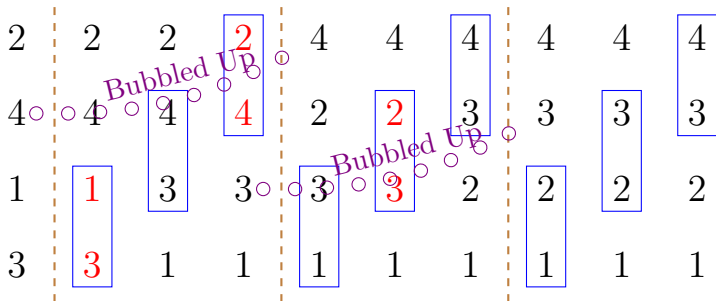
---

```
1: procedure BUBBLESORT( $A : a_1 \ a_2 \ \cdots \ a_n$ )
2:   repeat
3:     swapped  $\leftarrow$  false
4:     for  $i \leftarrow 1 : n - 1$  do
5:       if  $a_i > a_{i+1}$  then
6:         SWAP( $a_i, a_{i+1}$ )
7:         swapped  $\leftarrow$  true
8:   until no adjacent inversionsswapped = false
```

---

▷ Pick  $i$

# Bubble Sort: Example



# Bubble Sort: Finiteness

---

```

1: procedure BUBBLESORT( $A : a_1 \ a_2 \ \cdots \ a_n$ )
2:   repeat
3:     swapped  $\leftarrow$  false
4:     for  $i \leftarrow 1 : n - 1$  do            $\triangleright$  Pick  $i$ 
5:       if  $a_i > a_{i+1}$  then
6:         SWAP( $a_i, a_{i+1}$ )
7:         swapped  $\leftarrow$  true
8:   until swapped = false            $\triangleright$  No swaps

```

---

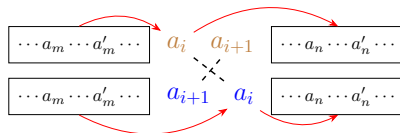
The inner “**for**” loops:

- 1)  $\exists$  loop : no swaps  $\implies$  swapped = false  $\implies$  terminates
- 2)  $\forall$  loop : has swaps      **Impossible!**

# Bubble Sort: Finiteness

Total #inversions is finite.

Effects of  $\text{SWAP}(a_i, a_{i+1})$  on #inversions<sup>1</sup>:



$-1 : (a_i, a_{i+1})$

$+0$

$\text{SWAP}(a_i, a_{i+1}) \implies -1 \text{ inversion}$

<sup>1</sup>Not on #adjacent inversions!

# Bubble Sort: Correctness

**Finiteness**  $\implies \exists \text{ loop : no swaps}$   
 $\implies A$  has no adjacent inversions any more  
 $\implies A$  is already sorted.

# Optimizing Bubble Sort (I)<sup>2</sup>

Idea: After each “**for**” loop, one more element is settled.

---

```

1: procedure BUBBLESORT( $A : a_1 \ a_2 \ \cdots \ a_n$ )
2:    $n \leftarrow \text{len}(A)$ 
3:   repeat
4:     swapped  $\leftarrow$  false
5:     for  $i \leftarrow 1 : n - 1$  do
6:       if  $a_i > a_{i+1}$  then
7:         SWAP( $a_i, a_{i+1}$ )
8:         swapped  $\leftarrow$  true
9:      $n \leftarrow n - 1$ 
10:  until swapped = false

```

---

▷ One maximal bubbles up

---

<sup>2</sup>See Appendix for “Optimizing Bubble Sort (II)”.

# Bubble Sort

- 1 Sorting
- 2 Bubble Sort
- 3 Analysis**



# Time Complexity of Bubble Sort

- Finiteness is NOT enough  $\implies$  Quantitative finiteness
- Time on real computers varies  $\implies$  #Ops on our model:

$|C|$  : #Comparisons (**if**  $a_i > a_{i+1}$ )

$|S|$  : #Swaps (SWAP( $a_i, a_{i+1}$ ))

- Different inputs  $\implies |C|$  and  $|S|$  vary:
  - Best-case, worst-case, and average-case analysis

# Best-case and Worst-case Analysis

Best-case: 1 2 3 4 5 6 7 8

Best-case:  
ascendingly sorted

Worst-case:  
descendingly sorted

$$|C| = (\min : n - 1, \quad \max : \frac{n^2 - n}{2});$$

$$|S| = (\min : 0, \quad \max : \frac{n^2 - n}{2}).$$

Worst-case: 8 7 6 5 4 3 2 1

$$\# \text{inversions} = (n - 1) + (n - 2) + \cdots + 1 = \frac{n^2 - n}{2}.$$

# $|S| : \text{\#Swaps (Average Analysis)}^3$

Assumptions on inputs:

1. The input is a random permutation
2. All numbers are distinct

$$\boxed{\text{SWAP}(a_i, a_{i+1}) \implies -1 \text{ inversion}}$$

$$|S| = \mathbb{E}(\text{\#inversions})$$

---

<sup>3</sup>The calculation of  $|C|$  in average is much more involved.

# $|S| : \# \text{Swaps}$ (Average Analysis)

$$I_{ij} = \begin{cases} 1 & (a_i, a_j) \text{ is an inversion} \\ 0 & \text{o.w.} \end{cases}$$

$$X = \sum_j \sum_{i < j} I_{ij} \quad (\# \text{inversions})$$

$$\mathbb{E}(X) = \mathbb{E}\left(\sum_j \sum_{i < j} I_{ij}\right) = \sum_j \sum_{i < j} \mathbb{E}(I_{ij})$$

$$\mathbb{E}(I_{ij}) = \mathbb{P}\{I_{ij} = 1\} = \frac{1}{2} \quad (a_i > a_j \vee a_i < a_j)$$

$$\mathbb{E}(X) = \sum_j \sum_{i < j} \frac{1}{2} = \binom{n}{2} \cdot \frac{1}{2} = \frac{n(n-1)}{4} = O(n^2)$$

# Faster Algorithms

*It took a good deal of work to analyze the bubble sort; and although [...], the results are disappointing since they tell us that **the bubble sort isn't really very good at all.***

— Donald E. Knuth

faster:  $O(n^2) \rightarrow O(n \lg n)$ ?

... and faster:  $O(n \lg n) \rightarrow O(n)$ ?





[hengxin0912@gmail.com](mailto:hengxin0912@gmail.com)



# Bubble Sort

## 4 Appendix

# Bubble Sort: Correctness

---

```

1: procedure BUBBLESORT( $A : a_1 \ a_2 \ \cdots \ a_n$ )
2:   repeat
3:     swapped  $\leftarrow$  false
4:     for  $i \leftarrow 1 : n - 1$  do       $\triangleright$  Loop invariant?
5:       if  $a_i > a_{i+1}$  then           $\triangleright$  CAS
6:         SWAP( $a_i, a_{i+1}$ )
7:       swapped  $\leftarrow$  true
8:   until swapped = false

```

---

## Loop invariant:

Before the  $k$ -th ( $k \geq 1$ ) “**for**” loop,  $a_{n-(k-1)} \cdots a_n$

- (1) consists of the largest  $(k - 1)$  elements
- (2) in sorted order.

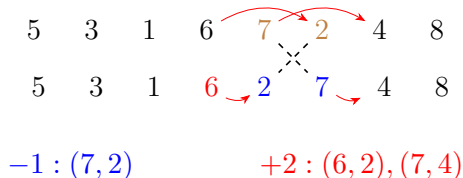
Correctness: Initialization + Maintenance + Termination



# Bubble Sort: Finiteness

Idea: well-founded relation over  $\mathbf{N}$

Effects of  $\text{SWAP}(a_i, a_{i+1})$  on adjacent inversions:



# Optimizing Bubble Sort (II)

Idea: After each “**for**” loop, all elements after “**lsp**” are settled.

---

```

1: procedure BUBBLESORT( $A : a_1 \ a_2 \ \cdots \ a_n$ )
2:   repeat
3:     swapped  $\leftarrow$  false
4:     lsp  $\leftarrow$  0                                 $\triangleright$  lsp: the last swap position
5:     for  $i \leftarrow 1 : n - 1$  do
6:       if  $a_i > a_{i+1}$  then
7:         SWAP( $a_i, a_{i+1}$ )
8:         swapped  $\leftarrow$  true
9:         lsp  $\leftarrow$  i                                 $\triangleright$  Update lsp
10:     $n \leftarrow$  lsp                                 $\triangleright$  Elements after lsp are sorted
11:  until swapped = false lsp = 0

```

---