



Bubble Sort Algorithm

Bubble Sort Algorithm

- The algorithm of bubble sort
 - Keep comparing a pair of items in the array back and forth.
 - If the first item is greater than the second item, it will swap.
 - The process of going through the array to compare and swap items from start to end is called a “pass”.
 - After each “pass”, if there any swap occur, the process starts all over again.
 - If there is no swap in a “pass”, then the array is sorted and the process can terminate.

Bubble Sort

- Array of numbers "5 1 4 2 8".
- Sort from lowest number to highest number using bubble sort algorithm.
- In each step, elements written in **bold** are being compared.
- (**5** **1** 4 2 8) (**1** **5** 4 2 8)
 - Compares the first two elements,
 - Swaps if first number is greater than second number.
 - And set swap to true. (swap: true)

Bubble Sort

- First Pass: swap: false
- (5 1 4 2 8)

Bubble Sort

- First Pass: 1 swap: true
- (5 1 4 2 8) -> (1 5 4 2 8) Swap since $5 > 1$

Bubble Sort

- First Pass: 2 swap: true
- $(5 \ 1 \ 4 \ 2 \ 8) \rightarrow (1 \ 5 \ 4 \ 2 \ 8)$
 $(1 \ 5 \ 4 \ 2 \ 8) \rightarrow (1 \ 4 \ 5 \ 2 \ 8)$ Swap since $5 > 4$

Bubble Sort

- First Pass: 3 swap: true
- $(5 \ 1 \ 4 \ 2 \ 8) \rightarrow (1 \ 5 \ 4 \ 2 \ 8)$.
 $(1 \ 5 \ 4 \ 2 \ 8) \rightarrow (1 \ 4 \ 5 \ 2 \ 8)$
 $(1 \ 4 \ 5 \ 2 \ 8) \rightarrow (1 \ 4 \ 2 \ 5 \ 8)$, Swap since $5 > 2$

Bubble Sort

- First Pass: 4 swap: true
- $(5 \ 1 \ 4 \ 2 \ 8) \rightarrow (1 \ 5 \ 4 \ 2 \ 8)$
- $(1 \ 5 \ 4 \ 2 \ 8) \rightarrow (1 \ 4 \ 5 \ 2 \ 8)$
- $(1 \ 4 \ 5 \ 2 \ 8) \rightarrow (1 \ 4 \ 2 \ 5 \ 8)$
- $(1 \ 4 \ 2 \ 5 \ 8) \rightarrow (1 \ 4 \ 2 \ 5 \ 8)$, Elements are already in order ($8 > 5$), do not swap.

Bubble Sort

- Second Pass: swap: false
- (1 4 2 5 8)

Bubble Sort

- Second Pass:
swap: false
- (**1 4** 2 5 8) (**1 4** 2 5 8) //do not swap

Bubble Sort

- Second Pass: 2 swap: true
- $(1\ 4\ 2\ 5\ 8)$ $(1\ 4\ 2\ 5\ 8)$
 $(1\ \textcolor{red}{4}\ 2\ 5\ 8)$ $(1\ \textcolor{red}{2}\ \textcolor{red}{4}\ 5\ 8)$, Swap since $4 > 2$ (swap occur here, need another pass)

Bubble Sort

- Second Pass: 3 swap: true
- (1 4 2 5 8) (1 4 2 5 8)
- (1 4 2 5 8) (1 2 4 5 8)
- (1 2 4 5 8) (1 2 4 5 8) //Do not swap

Bubble Sort

- **Second Pass:4** swap: true
- (1 4 2 5 8) (1 4 2 5 8)
(1 4 2 5 8) (1 2 4 5 8)
(1 2 4 5 8) (1 2 4 5 8)
(1 2 4 **5 8**) (1 2 4 **5 8**) //Do not swap
- Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one **whole** pass without **any** swap to know it is sorted.

Bubble Sort

- Third Pass: swap: false
- (1 2 4 5 8)

Bubble Sort

- Third Pass:
swap: false
- (1 2 4 5 8)(1 2 4 5 8)//Do not swap

Bubble Sort

- Third Pass:2 swap: false
- $(1\ 2\ 4\ 5\ 8)(1\ 2\ 4\ 5\ 8)$
 $(1\ \textcolor{red}{2}\ \textcolor{red}{4}\ 5\ 8)(1\ \textcolor{red}{2}\ \textcolor{red}{4}\ 5\ 8)$ //Do not swap

Bubble Sort

- Third Pass:3 swap: false
- (1 2 4 5 8) (1 2 4 5 8)
- (1 2 4 5 8) (1 2 4 5 8)
- (1 2 **4 5** 8) (1 2 **4 5** 8) //Do not swap

Bubble Sort

- Third Pass:4 swap: false
- (1 2 4 5 8) (1 2 4 5 8)
- (1 2 4 5 8) (1 2 4 5 8)
- (1 2 4 5 8) (1 2 4 5 8)
- (1 2 4 **5 8**) (1 2 4 **5 8**) (no swap occur in this pass, so the array is sorted)
Finally, the array is sorted, and the sorting process terminate.

Exercise

- Given the number array below:
 - (5,3,4)
 - Work out the bubble sort swap sequence for each pass
 - Review your answer in the next slide.

Exercise Review

- Pass 1
 - (5,3,4) -> (3,5,4) swap: true
 - (3,5,4) -> (3,4,5) swap: true
- Pass 2
 - (3,4,5) -> (3,4,5) swap: false
 - (3,4,5) -> (3,4,5) swap: false