

CMSC 132: OBJECT-ORIENTED PROGRAMMING II



Insertion Sort and the clone method

Department of Computer Science
University of Maryland, College Park

Insertion Sort

Concept: Insertion Sort builds a sorted section of the array one element at a time by inserting each new element into its correct position.

Analogy: Sorting a hand of playing cards—pick up one card at a time and insert it in the right place.

Process: Assume the first element is already sorted.

1. Take the next element and compare it with elements in the sorted part.
2. Shift elements if necessary to make space for the new element.
3. Repeat until the array is fully sorted.

Insertion Sort: Example

- **Sorting [5, 3, 8, 4, 2]**
- **Pass 1 (Insert 3 into [5])**
 - (5,3) → shift → [5, 5, 8, 4, 2]
 - Insert 3 → [3, 5, 8, 4, 2]
- **Pass 2 (Insert 8 into [3, 5])**
 - No shifts needed → [3, 5, 8, 4, 2]
- **Pass 3 (Insert 4 into [3, 5, 8])**
 - (8,4) → shift → [3, 5, 8, 8, 2]
 - (5,4) → shift → [3, 5, 5, 8, 2]
 - Insert 4 → [3, 4, 5, 8, 2]
- **Pass 4 (Insert 2 into [3, 4, 5, 8])**
 - (8,2) → shift → [3, 4, 5, 8, 8]
 - (5,2) → shift → [3, 4, 5, 5, 8]
 - (4,2) → shift → [3, 4, 4, 5, 8]
 - (3,2) → shift → [3, 3, 4, 5, 8]
 - Insert 2 → [2, 3, 4, 5, 8]

Note: Starting at the end of sorted part to find insertion point combines searching and shifting into one pass!

- See: `InsertionSort`

Insertion Sort

- **Worst-case ($O(n^2)$):** When the array is sorted in reverse order, every element must be compared and shifted for every insertion.
- This results in $O(n^2)$ time complexity.
- **Best-case ($O(n)$):** When the array is already sorted, each element is just compared once, leading to $O(n)$ complexity
- **Average-case ($O(n^2)$):** On average, elements are inserted in the middle of the sorted section, leading to $O(n^2)$ performance.
- **Space Complexity ($O(1)$):** Uses only a constant amount of extra space (in-place sorting).

The clone Method in Java

- clone() is a method from Object that creates a copy of an object.
- Defined in java.lang.Object:

```
protected Object clone() throws CloneNotSupportedException
```

- **Access Modifiers:** Protected → Subclasses must override it to make it public.
- **Returns:** A new object that is a copy of the original.
- **Throws:** CloneNotSupportedException if the class does not implement Cloneable.

The Cloneable Marker Interface & Shallow Copy

- A **marker interface** in Java is an interface that **does not declare any methods or fields** but is used to signal a special property or behavior to the Java runtime or other code. It acts as a **tag** that allows objects of a class implementing it to be treated differently.
- Cloneable is a marker interface: No methods, just a flag for clone() to work.
- See: <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Cloneable.html>
- If a class does not implement Cloneable, clone() throws CloneNotSupportedException.
See: MyClass
- **Shallow Copy by Default:** Fields are copied as-is.
- If a field is a reference (e.g., array, object), the reference is copied, **not** the object itself.
- **See: The Cloning package with Mouse first, then Computer, and finally SuperComputer.**