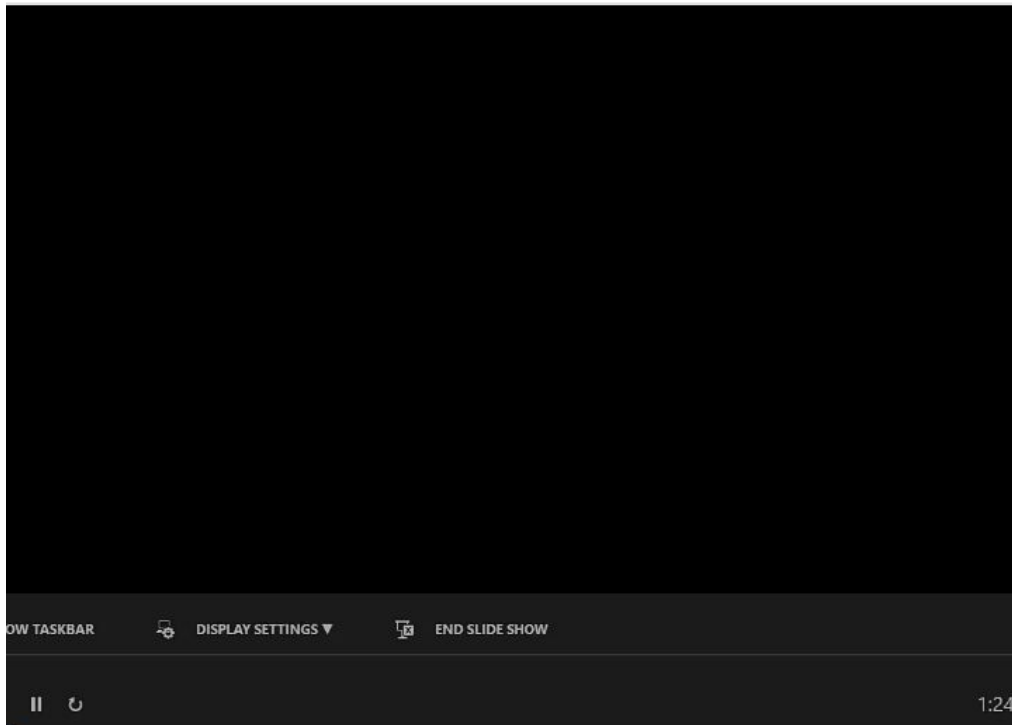




aaup.zoom.us/re



## Sorting Problem

Sorting algorithms can be categorized into several categories based on their approach to sorting the elements of an array or a list. Here are the main categories of sorting algorithms:

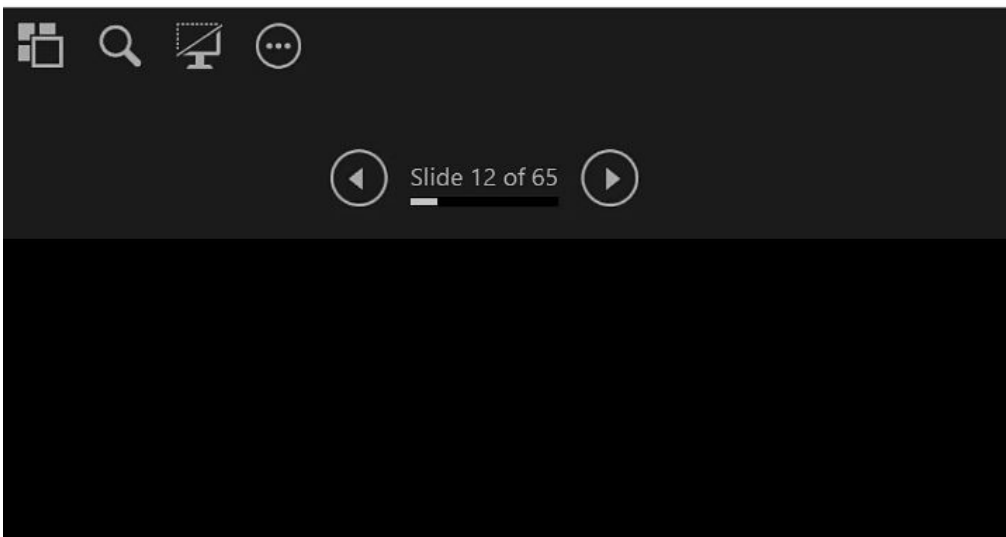
### 4. Stable sorting algorithms:

- These algorithms maintain the relative order of elements with equal keys. This means that if two elements have the same value, they will appear in the same order in the sorted array as they did in the original unsorted array.
- Examples of stable sorting algorithms include [insertion sort](#), [merge sort](#), and [counting sort](#).

### 5. Unstable sorting algorithms:

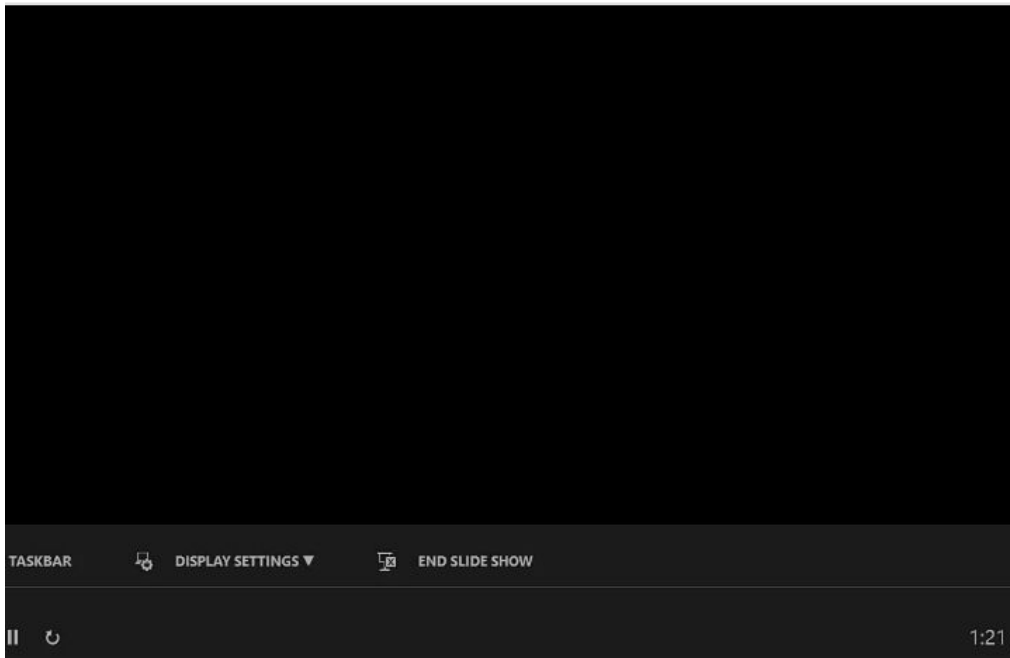
- These algorithms do not guarantee that the relative order of elements with equal keys will be maintained.
- Examples of unstable sorting algorithms include [quicksort](#) and [heap sort](#).

4 3 ?





aaup.zoom.us/re



## Sorting Problem

Sorting algorithms can be categorized into several categories based on their approach to sorting the elements of an array or a list. Here are the main categories of sorting algorithms:

### 1. Comparison-based sorting algorithms

- These algorithms compare elements of the array to be sorted using a comparison operator, such as less than or greater than.
- Examples of comparison-based sorting algorithms include [bubble sort](#), [selection sort](#), [insertion sort](#), [merge sort](#), [quicksort](#), and [heap sort](#).

### 2. Non-comparison-based sorting algorithms

- These algorithms do not rely on comparison operators to sort the elements of an array. Instead, they use specialized techniques to sort the elements more efficiently.
- Examples of non-comparison-based sorting algorithms include [counting sort](#), [radix sort](#), and [bucket sort](#).

### 3. In-place sorting algorithms

- These algorithms do not require any additional memory to sort the elements of an array. They rearrange the elements within the original array itself.
- Examples of in-place sorting algorithms include [selection sort](#), [bubble sort](#), [insertion sort](#), [quicksort](#), and [heap sort](#).

### 4. "Out-of-place" or "External" sorting algorithms

- These algorithms work by dividing the input data into smaller parts that can fit into memory, sorting each part separately, and then merging the sorted parts into a single sorted output.
- Out-of-place sorting algorithms are typically used when the data to be sorted is too large to fit into memory all at once, or when the input is stored on external storage such as a hard disk or tape drive.

11

