

✓ - What is comparator

✓ → how it works

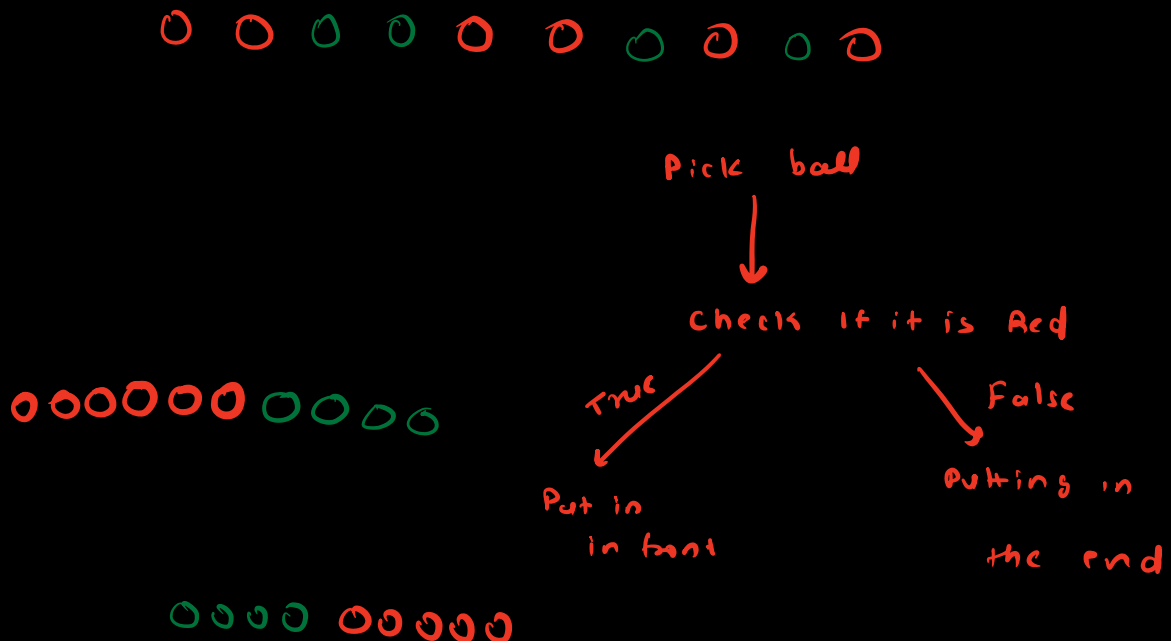
✓ → Implementation

→ C++

→ Java

✓ → Sort based on Tenth place

Comparator tells you decide the custom parameter for sorting.



Implementation

1. C++

sorting in ascending order

```
int main()
{
    ...
    Sort(arr, begin(), arr.end())
    ...
}
```

data type in array

boolean comparez (int a, int b)

```
{
    Return True    a = b, a < b
    Return False   b < a
}
```

```
int main()
{
    ...
    Sort(arr, begin(), arr.end(), comparez)
    ...
}
```

method name

Java

ascending order by default.

Collections.sort(A)

datatype in array

```
CompClassxyz implements Comparator<Int> {  
    @Override  
    int compare (Int a, Int b)  
    {  
        return 0    a == b  
        return -ve  a comes before b  
        return +ve  b comes before a  
    }  
}
```

object

Collections.sort(A, new CompClassxyz())
 ↓
 Arrays.sort

Q. Sort in increasing order of tens place

If number < 10, assume tens digit = 0

If 2 numbers have same tens digit, larger number comes first.

$$A = \{ \overset{1}{\uparrow} 15, \overset{1}{\uparrow} 11, \overset{0}{\uparrow} 7, \overset{1}{\uparrow} 19, \overset{2}{\uparrow} 24, \overset{2}{\uparrow} 22, \overset{8}{\uparrow} 89, \overset{1}{\uparrow} 312, \overset{6}{\uparrow} 4363 \}$$



$$ans = \{ 7, 312, 19, 15, 11, 24, 22, 4363, 89 \}$$

`CompClass xyz implements Comparator<Int> {`

`@Override`

`int compare (Int a, Int b)`

`int a1 = (a/10) % 10`

`int b1 = (b/10) % 10`

`if (a1 < b1) return -1`
`if (a1 > b1) return +1`

`→ if (a1 != b1) return a1 - b1`

`if (a > b) return -1`
`if (b > a) return +1`
`if (a == b) return 0`

`→ return b - a`

`}`

```
Public Class Solution {
```

```
    Public — solve ( int[] A ) {
```

```
        Sort ( A, new CompClassxyz () );
```

any name

```
    }
```

Tc for sorting

$$O(n \log n * \text{Tc of comp})$$

↓
O(1)