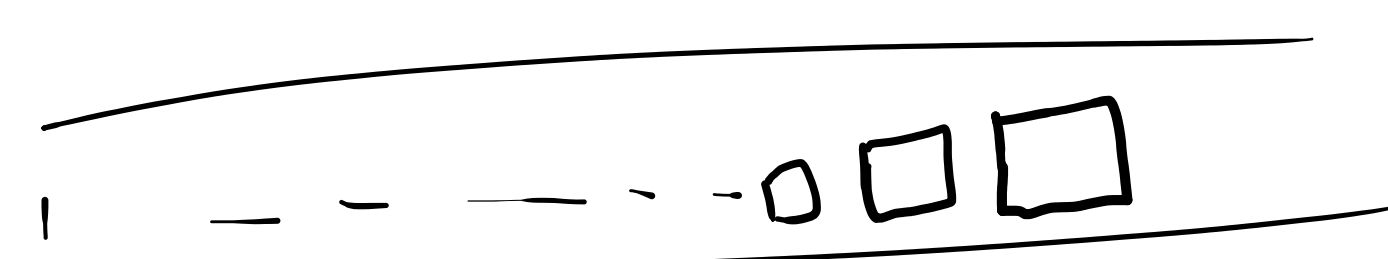


n data

Sorting

3-way

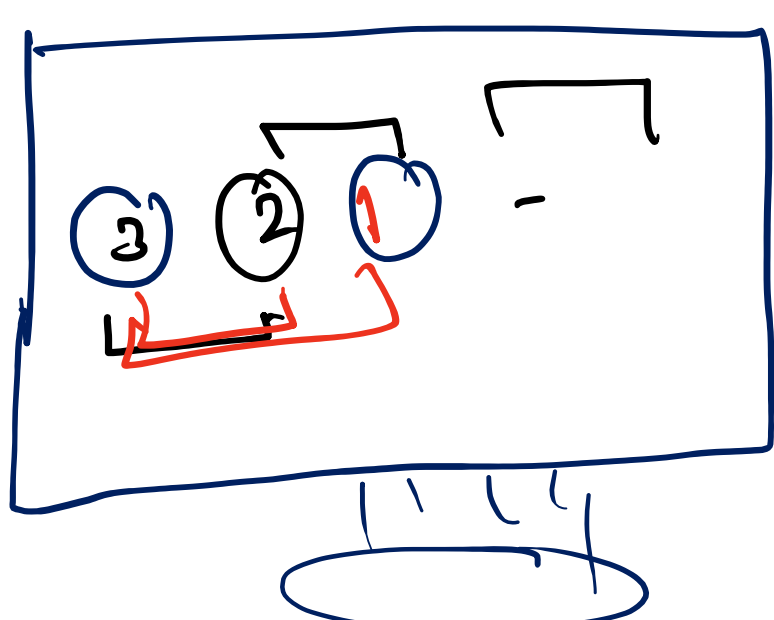


① 0 1 1 1 1 1

①) start line from start of

bubble
selection
insertion
counting
cycle
DMF
merge
quick
heap

① bubble sort



4 5

arr[i] > arr[i+1]

① 4 5 3 2 1
→ 4 5 3 2 1
4 3 5 2 1
4 3 2 5 1
4 3 2 1 5

② 4 3 2 1 5
→ 3 4 2 1 5
3 2 4 1 5
3 2 1 4 5

③ 3 2 1 4 5
→ 2 3 1 4 5
2 1 3 4 5

④ 2 1 3 4 5
→ 1 2 3 4 5

```
public static void Sort(int[] arr) {
    for (int turn = 1; turn < arr.length; turn++) {
        for (int i = 0; i < arr.length - turn; i++) {
            if (arr[i] > arr[i+1]) {
                int temp = arr[i];
                arr[i] = arr[i+1];
                arr[i+1] = temp;
            }
        }
    }
}
```

① 4 5 3 2 1
i=0 4 5 3 2 1
i=1 4 3 5 2 1
i=2 4 3 2 5 1
i=3 4 3 2 1 5
② 3 2 1 4 5
i=0 3 2 1 4 5
i=1 2 3 1 4 5
i=2 2 1 3 4 5
i=3 2 1 3 4 5

0 1 2 3 4 5 7
4, -1, 5, 3, 2, 1, 7
↑ ↑ ↑ ↑ ↑

mini = i

for (j=i+1; j<arr.length; j++) {
if (arr[j] < arr[mini]) {
mini=j;
}

3 < 5
2 < 3
2 < 1

4, -1, 5, 3, 2, 1, 7
i=0 0 1 2 3 4 5 6
-1 4 5 3 2 1 7
i=1 -1 1 5 3 2 4 7
i=2 -1 1 2 3 5 4 7
i=3 -1 1 2 3 5 4 7
i=4 -1 1 2 3 4 5 7
i=5 -1 1 2 3 4 5 7

① -comp
② (4, 2)
③ (3, 3)
④ (4, 5)
⑤ (5, 5)

item = arr[i]

j = i-1

while (j >= 0 && arr[j] > item) {
arr[j+1] = arr[j];
arr[j] = item;
j--;

0 1 2 3 4 5 6
1, 2, 7, 8, 9, 11, 4
j=5 1 2 7 8 9 4 11
j=4 1 2 7 8 4 9 11
j=3 1 2 7 4 8 9 11
j=2 1 2 4 7 8 9 11
j=1 1 4 2 7 8 9 11
j=0 -4 1 2 7 8 9 11
j=-1

[5, 3, -7, 4, -8, 2]

[5, 3, -7, 4, -8, 2]

[5 | 3 | -7 | 4 | -8 | 2]

3 5 -7 4 8 2
-7 3 5 4 -8 2
-7 3 4 5 -8 2
-8 -7 3 4 5 2
-8 -7 2 3 4 5