

MERGE K SORTED ARRAYS USING HEAPS

MERGE K SORTED ARRAYS USING HEAPS

SAY THE LISTS TO BE MERGED ARE
SORTED IN ASCENDING ORDER

GET THE **MINIMUM ELEMENT**
FROM THE HEAP - IT WILL BE THE
SMALLEST OF ALL FIRST
ELEMENTS IN THE K LISTS

KEEP TRACK OF WHICH LIST THIS
SMALLEST ELEMENT
ORIGINALLY CAME FROM

REMOVE THE **SMALLEST**
ELEMENT FROM EACH LIST AND
ADD IT TO THE MINIMUM HEAP -
THERE WILL BE K ELEMENTS IN
THE HEAP

GET THE NEXT SMALLEST
ELEMENT FROM THE **SAME** LIST

CONTINUE THIS TILL THE SORTED
LIST IS COMPLETELY FILLED

MERGE K SORTED ARRAYS USING HEAPS

SET UP A HEAP WITH CAPACITY 3

5 6 7 9 10 4

2 3 13 15 17 20 B

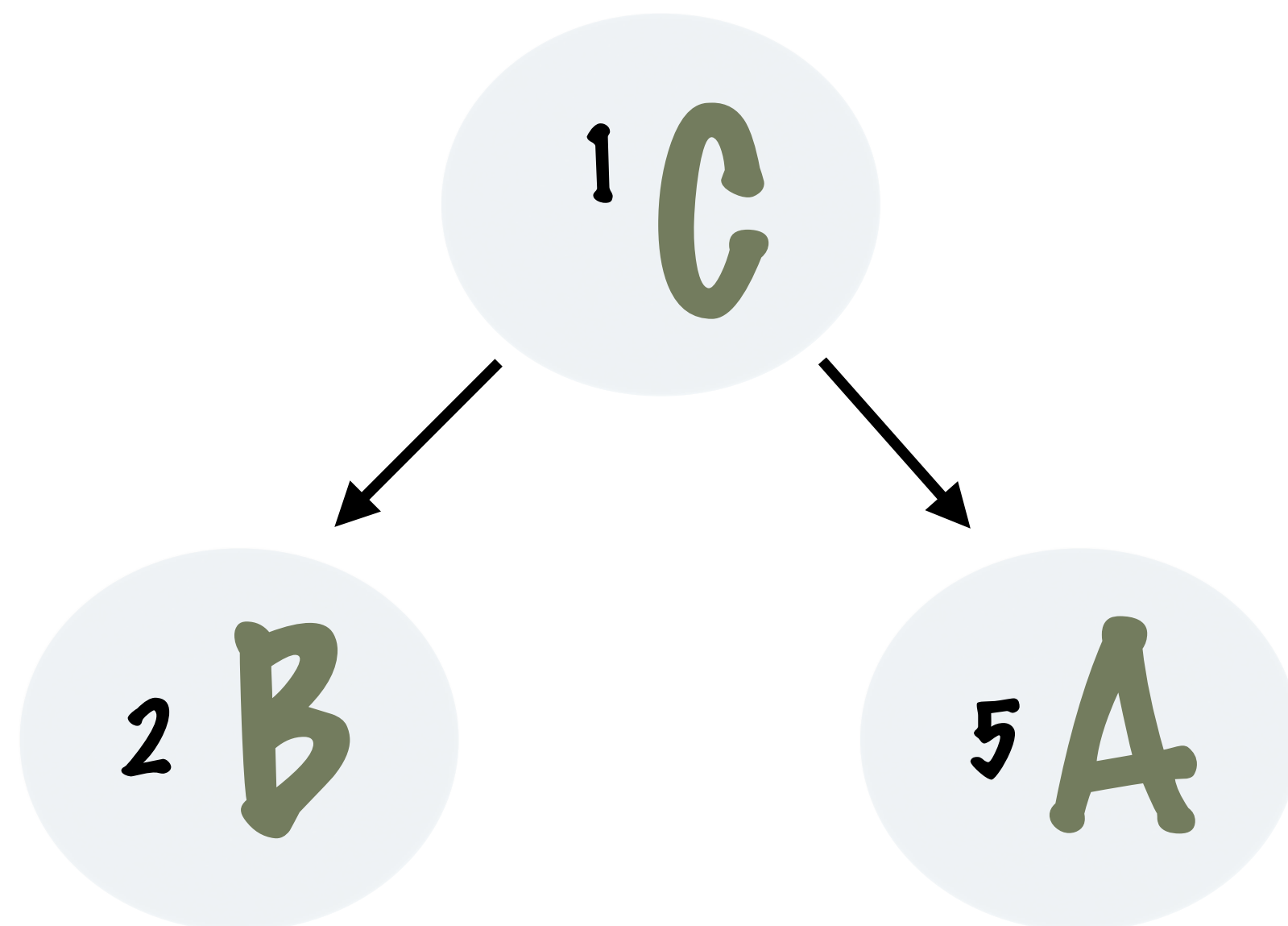
1 8 C

THE SORTED ARRAY

[illegible]

MERGE K SORTED ARRAYS USING HEAPS

REMOVE THE MINIMUM ELEMENT FROM THE HEAP



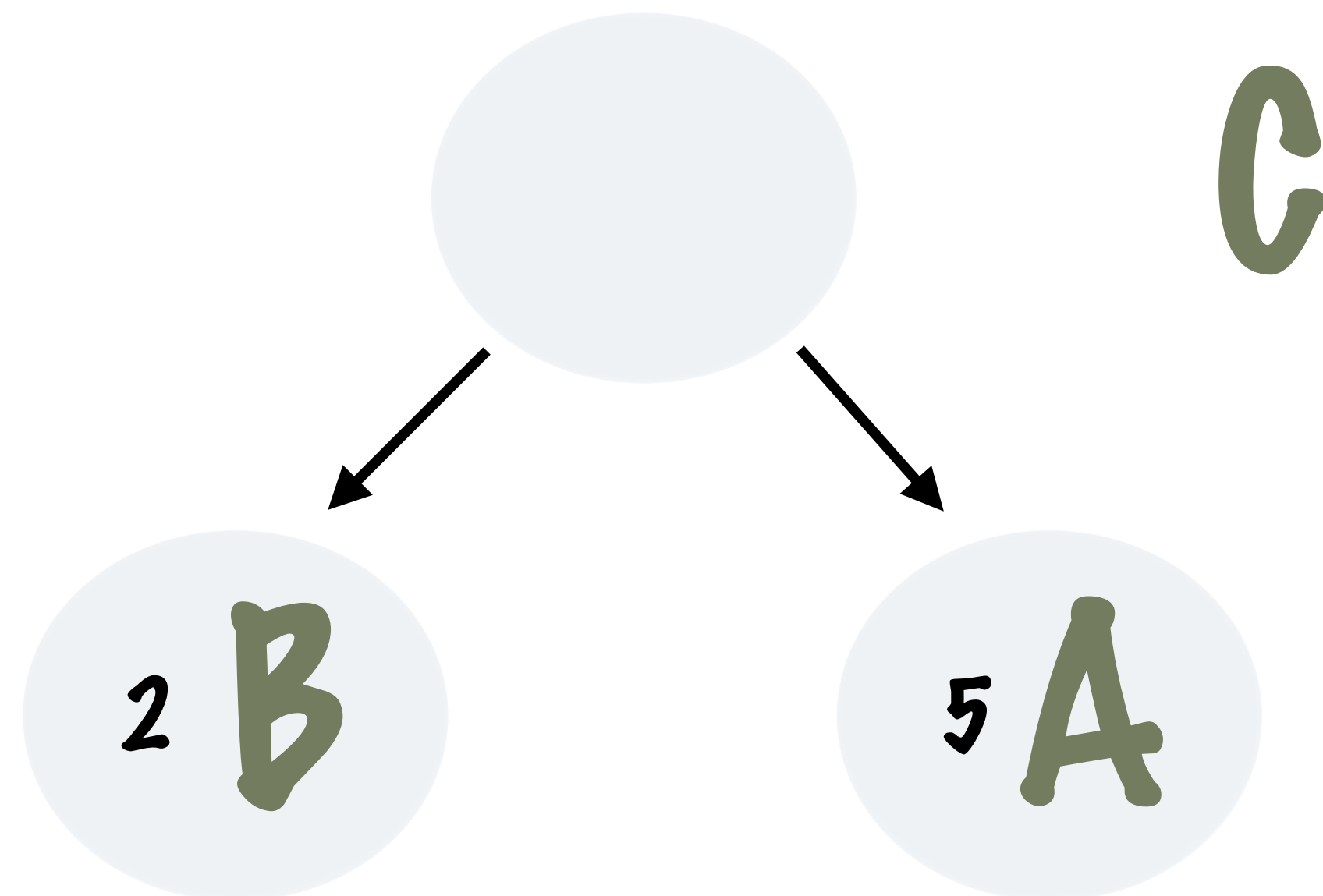
6	7	9	10	A	
3	13	15	17	20	B
8	C				

THE SORTED ARRAY

[illegible]

MERGE K SORTED ARRAYS USING HEAPS

GET THE NEXT ELEMENT FROM LIST C



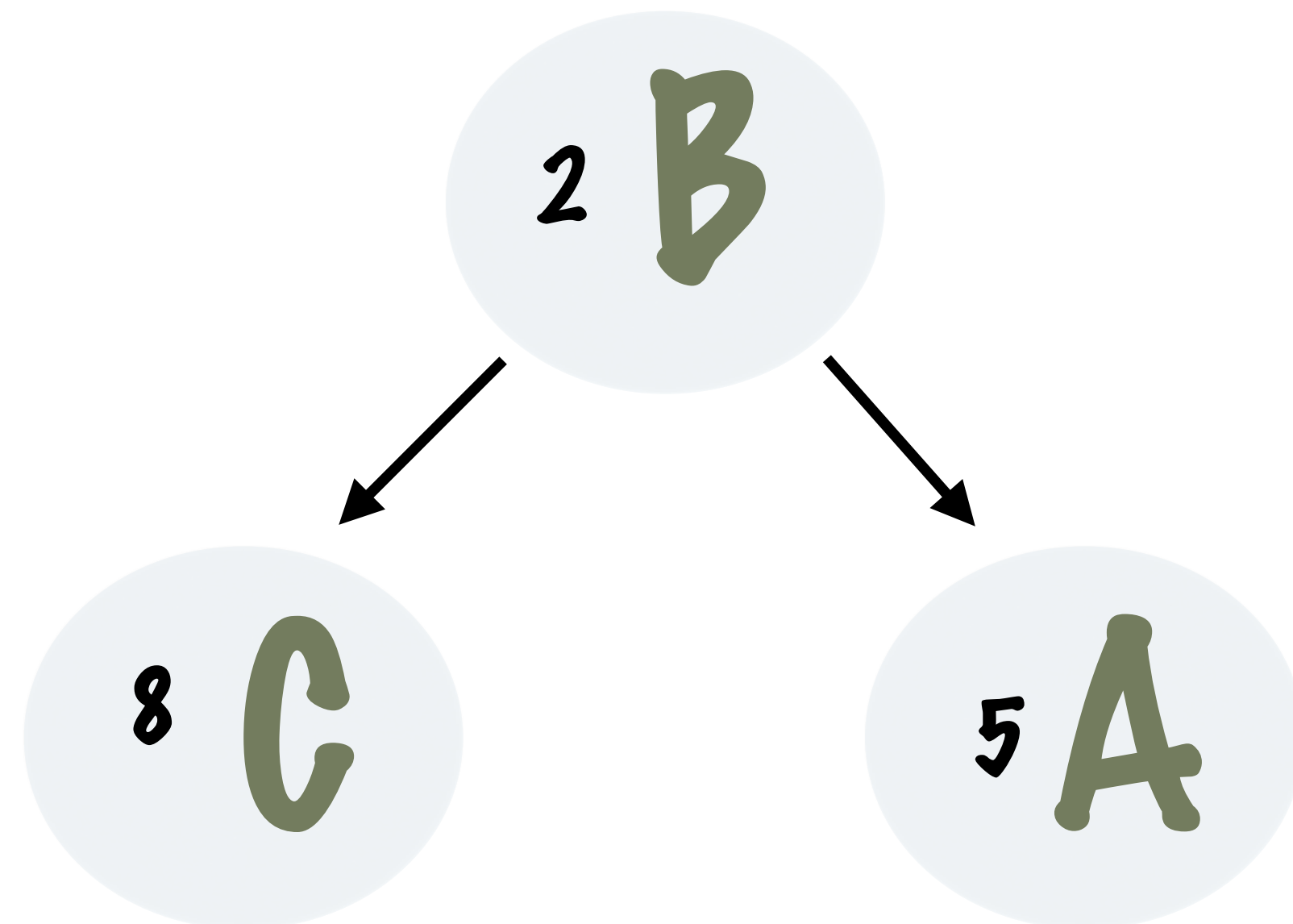
6	7	9	10	A	
3	13	15	17	20	B
8	C				

THE SORTED ARRAY

[illegible]

MERGE K SORTED ARRAYS USING HEAPS

REMOVE THE MINIMUM ELEMENT FROM THE HEAP



6	7	9	10	A	
3	13	15	17	20	B
C					

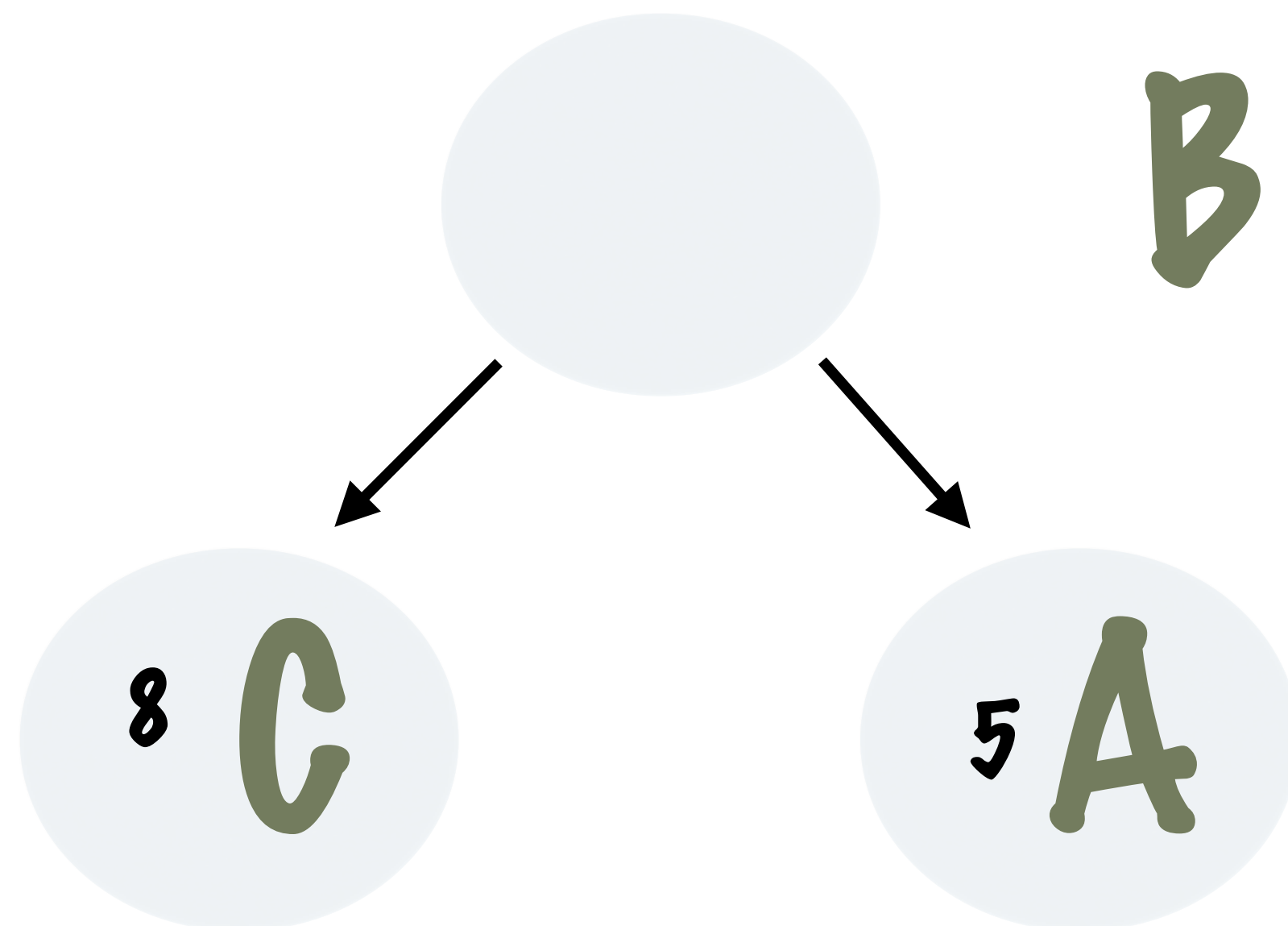
THE SORTED ARRAY

[illegible]

MERGE K SORTED ARRAYS USING HEAPS

GET THE NEXT ELEMENT FROM LIST B

6	7	9	10	A	
3	13	15	17	20	B
C					

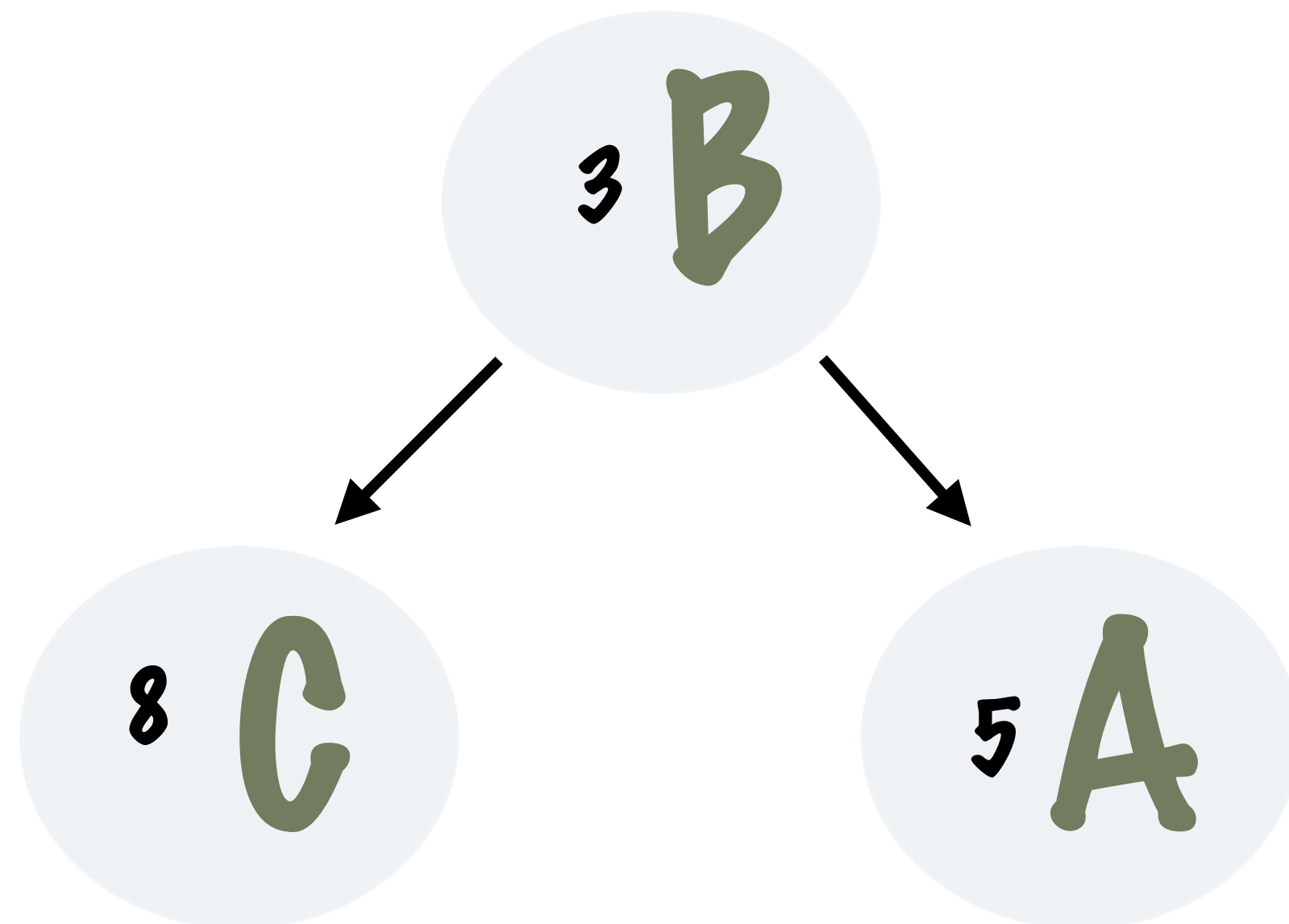


THE SORTED ARRAY

[illegible]

MERGE K SORTED ARRAYS USING HEAPS

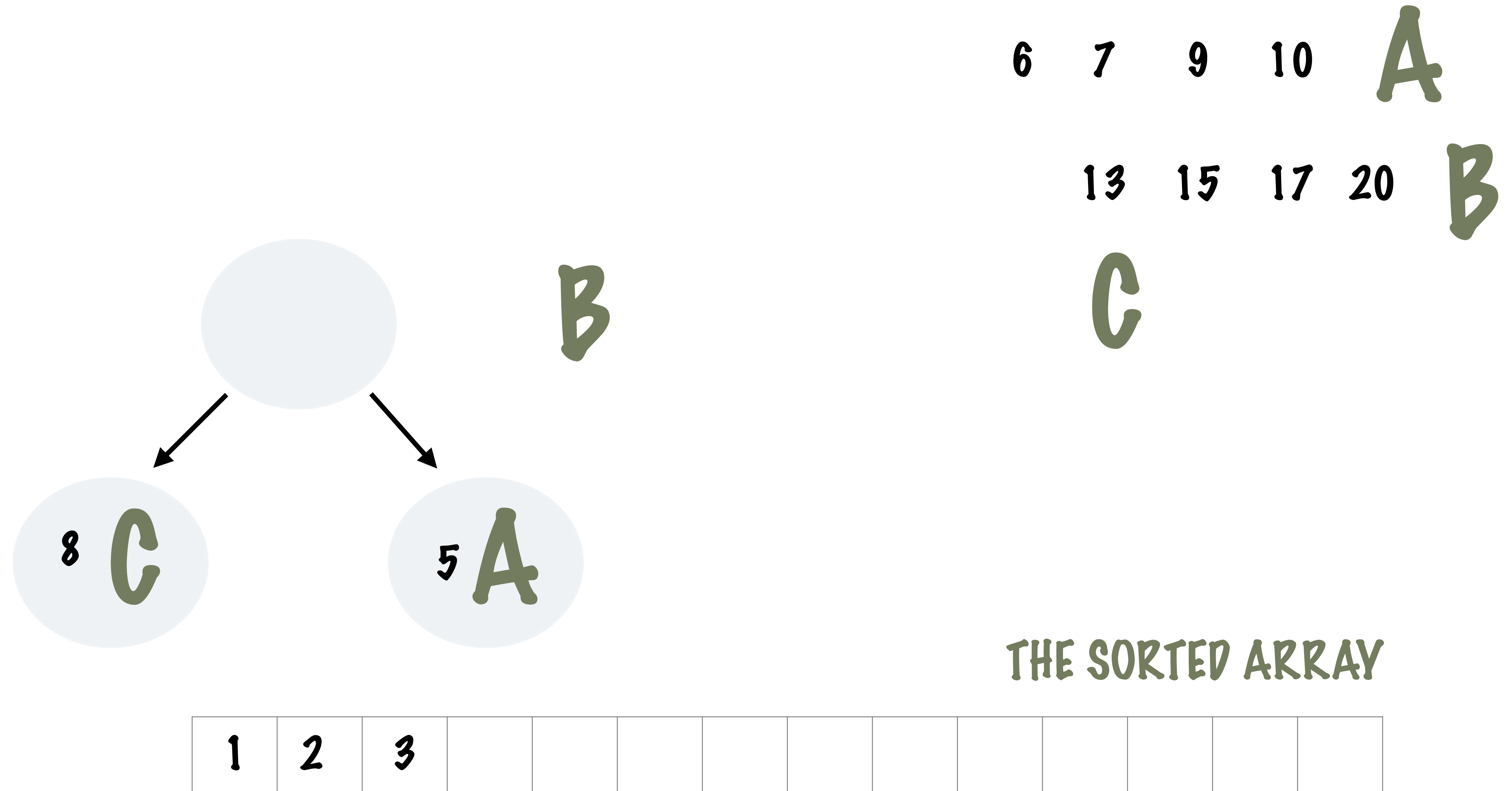
6	7	9	10	A
13	15	17	20	B
C				



THE SORTED ARRAY

[illegible]

MERGE K SORTED ARRAYS USING HEAPS



MERGE K SORTED ARRAYS USING HEAPS



THE SORTED ARRAY

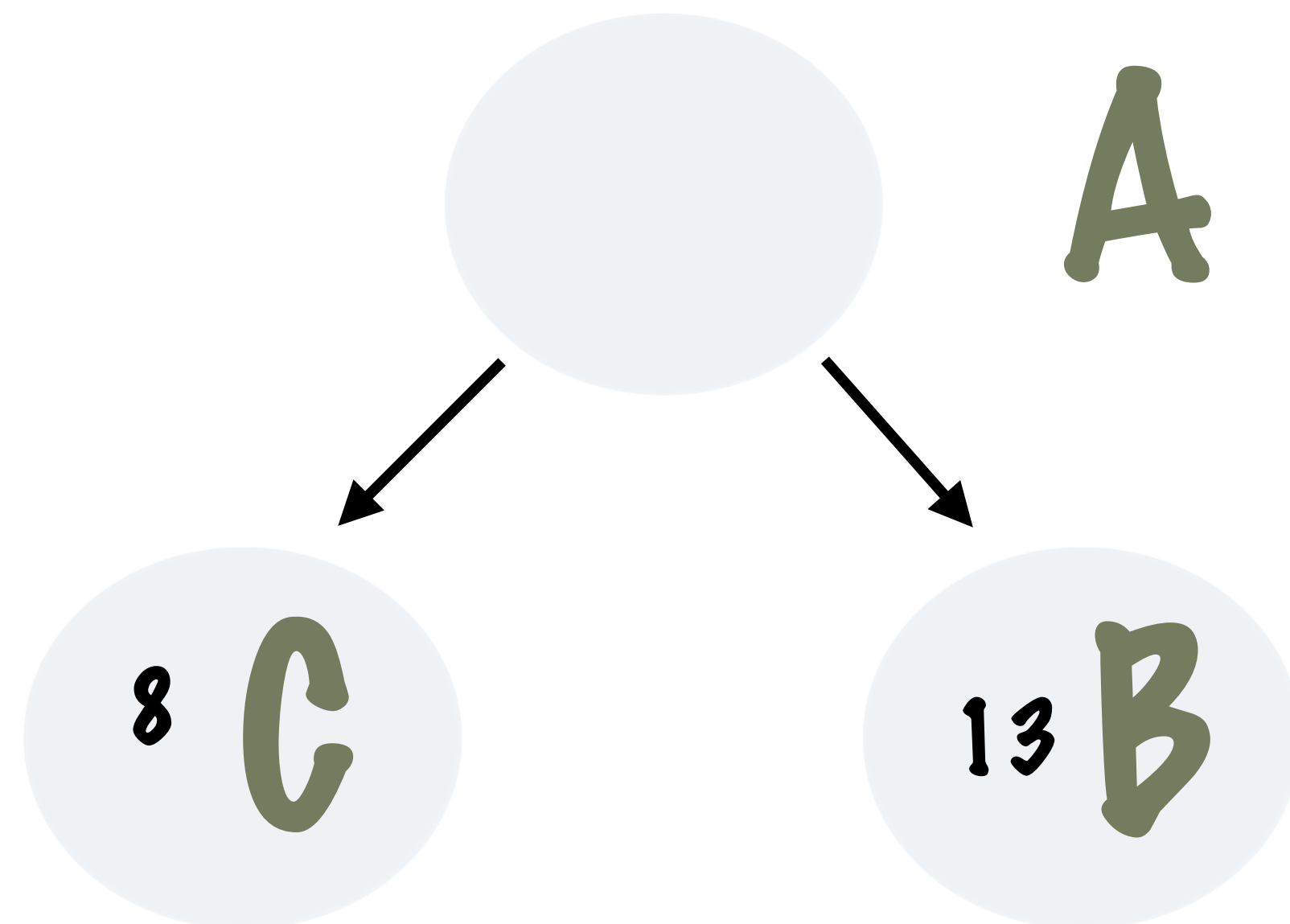
[illegible]

MERGE K SORTED ARRAYS USING HEAPS

6 7 9 10 A

15 17 20 B

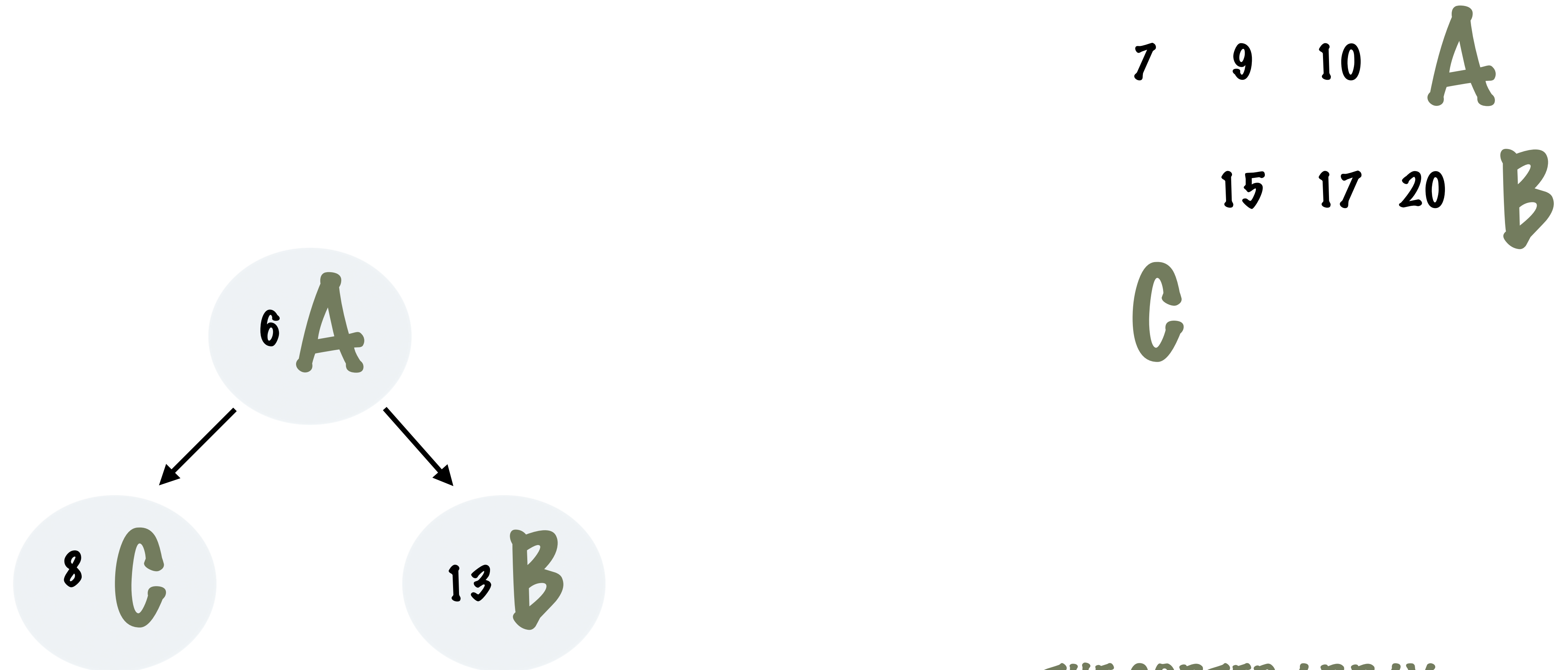
C



THE SORTED ARRAY

[illegible]

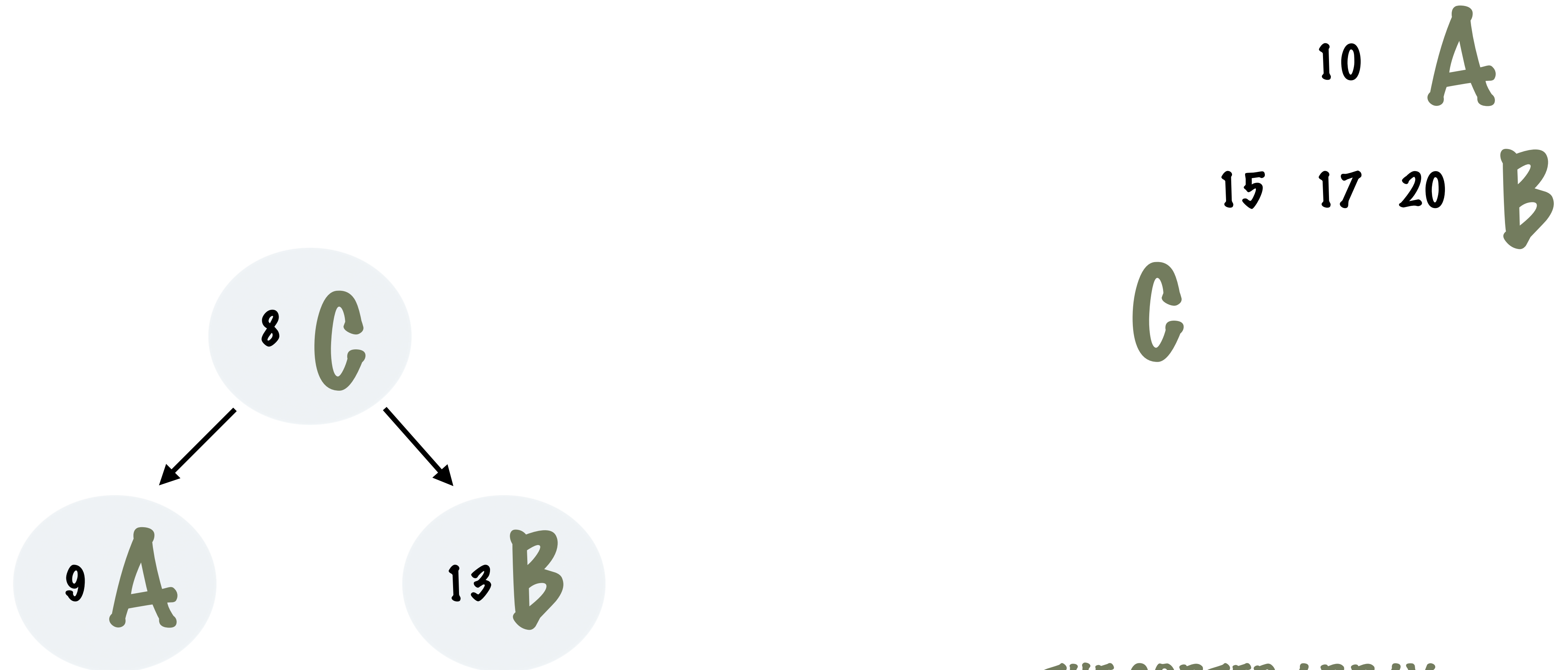
MERGE K SORTED ARRAYS USING HEAPS



THE SORTED ARRAY

[illegible]

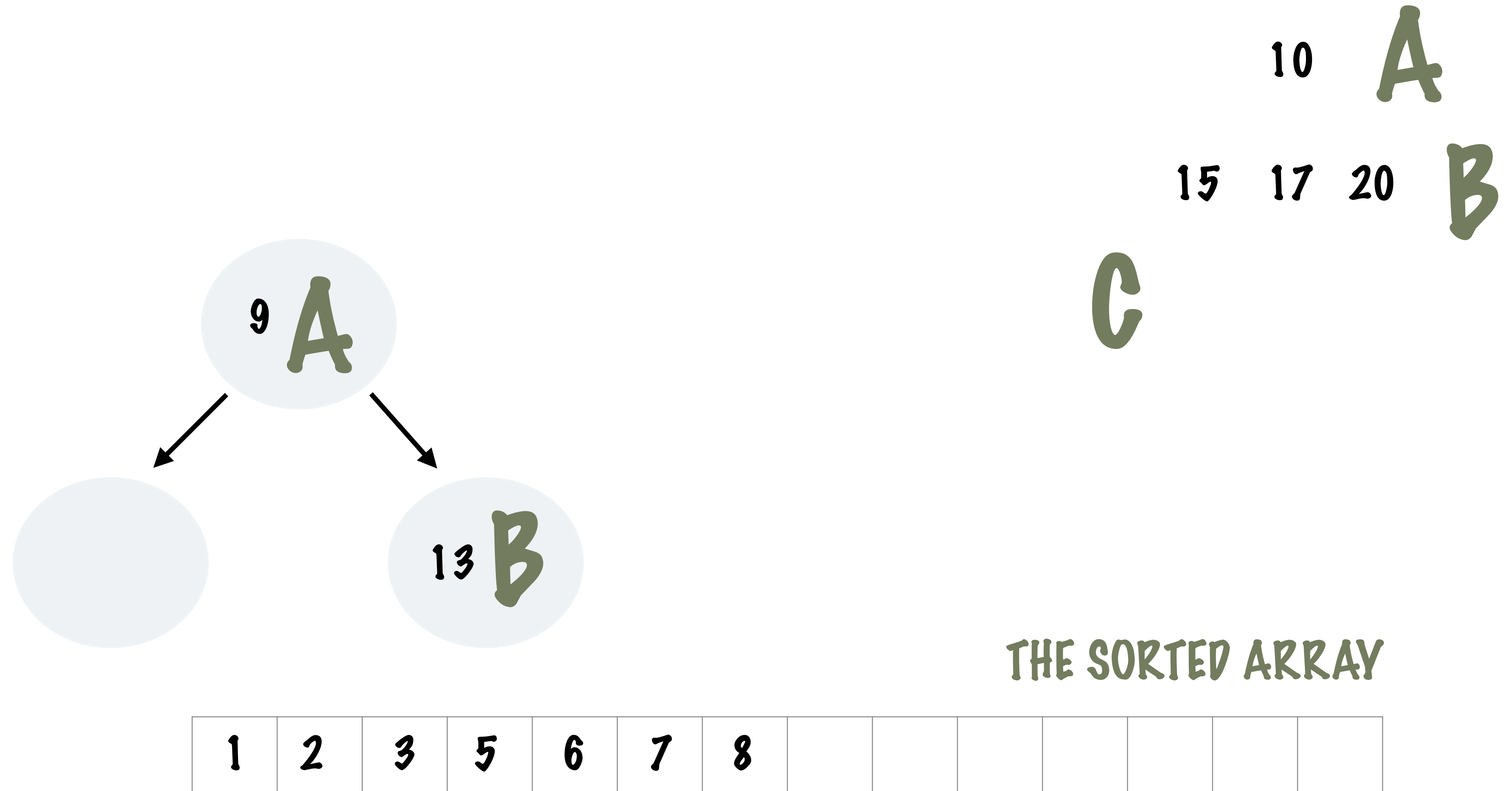
MERGE K SORTED ARRAYS USING HEAPS



THE SORTED ARRAY

1	2	3	5	6	7								
---	---	---	---	---	---	--	--	--	--	--	--	--	--

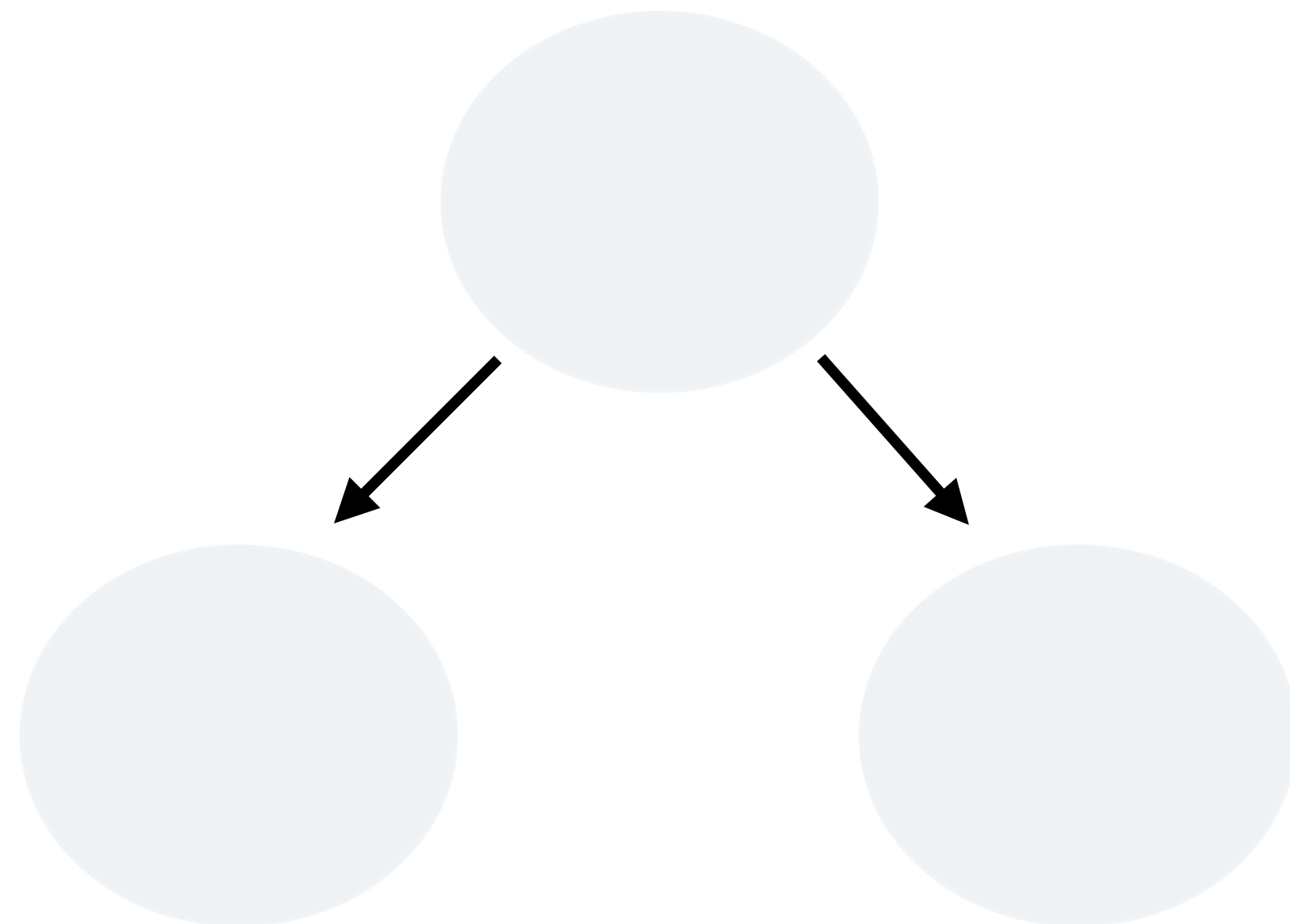
MERGE K SORTED ARRAYS USING HEAPS



MERGE K SORTED ARRAYS USING HEAPS

A
B

C



THE SORTED ARRAY

1	2	3	5	6	7	8	9	10	13	15	17	20	
---	---	---	---	---	---	---	---	----	----	----	----	----	--

A STRUCTURE HOLDING THE VALUE AND THE LIST IT CAME FROM

THE ELEMENT CAN BE COMPARED
USING THE VALUE IT HOLDS

```
public static class Element implements Comparable<Element> {
```

```
    private Integer listIndex;  
    private Integer value;
```

```
    public Element(Integer listIndex, Integer value) {  
        this.listIndex = listIndex;  
        this.value = value;  
    }
```

```
    public Integer getValue() {  
        return value;  
    }
```

```
    public Integer getListIndex() {  
        return listIndex;  
    }
```

```
    @Override  
    public int compareTo(Element element) {  
        return value - element.value;  
    }
```

```
}
```

THE INDEX OF THE LIST WHICH
ORIGINALLY HELD THIS ELEMENT

THE VALUE OF THE ELEMENT, THE
ACTUAL VALUES WHICH HAVE TO
BE ADDED TO THE SORTED LIST

MERGE K SORTED LISTS

```
public static void mergeKSortedLists(int totalElements, List<Integer>... lists)
    throws MinHeap.HeapFullException, MinHeap.HeapEmptyException {
    MinHeap<Element> minHeap = new MinHeap<>(Element.class, lists.length);

    List<Integer> sortedList = new ArrayList<>();
    for (int i = 0; i < lists.length; i++) {
        List<Integer> list = lists[i];
        if (!list.isEmpty()) {
            minHeap.insert(new Element(i, list.remove(0)));
        }
    }

    while (sortedList.size() < totalElements) {
        Element element = minHeap.removeHighestPriority();
        sortedList.add(element.getValue());

        List<Integer> list = lists[element.getListIndex()];
        if (!list.isEmpty()) {
            minHeap.insert(new Element(element.getListIndex(), list.remove(0)));
        }
    }

    printList(sortedList);
}
```

ARGUMENTS ARE AN ARRAY OF LISTS AND THE TOTAL NUMBER OF ELEMENTS TO BE PROCESSED

A MIN HEAP WITH CAPACITY EQUAL TO THE NUMBER OF LISTS TO MERGE

POPULATE THE MIN HEAP WITH THE SMALLEST ELEMENT FROM EVERY LIST

ADD THE MINIMUM TO THE FINAL SORTED ARRAY

GET THE NEXT ELEMENT FROM THE LIST WHICH HELD THE MINIMUM ELEMENT