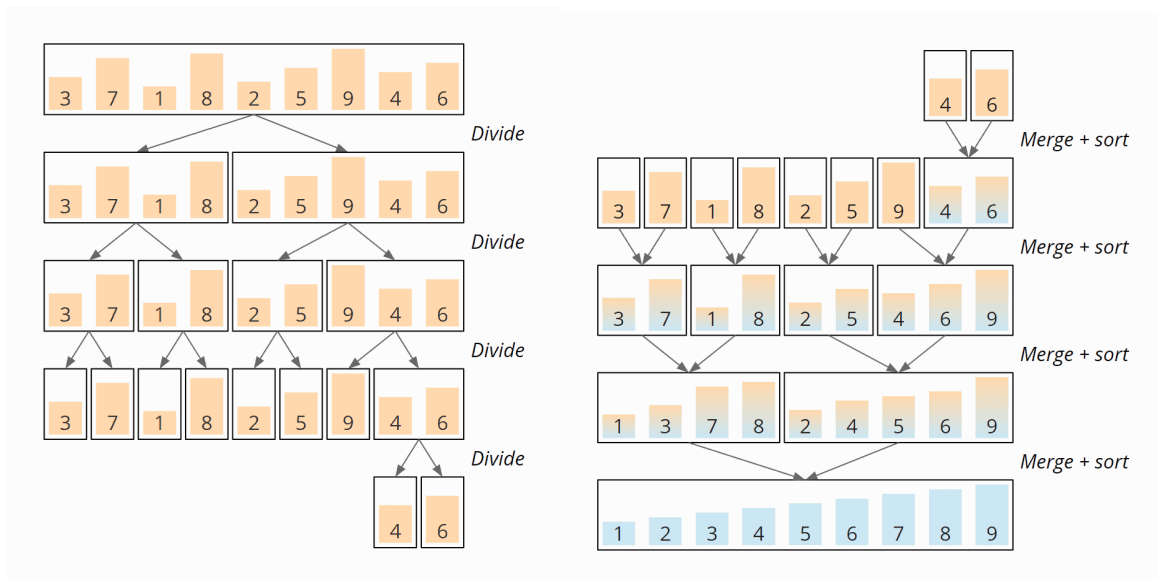


[A08] MergeSort

Grafische Beschreibung



Textuelle Beschreibung

MergeSort arbeitet nach dem Prinzip Divide and Conquer. Dem Algorithmus wird hierbei ein Array übergeben, welches so lange in 2 Hälften geteilt wird, bis nur mehr Arrays mit Länge 1 vorhanden sind.

Nach der Aufteilung werden die Arrays paarweise und sortiert wieder zusammengesetzt. Das passiert so lange bis am Ende wieder ein Array zusammengesetzt ist.

Laufzeit

- `public void sort(Person[] personen) -> $O(n \cdot \log(n))$`
- `public void merge(Person[] pers1, Person[] pers2, Person[] result, int start) -> $O(n \cdot \log(n))$`

Wichtige Elemente

```
public void merge(Person[] pers1, Person[] pers2, Person[] result, int start) {

    int pointerP1 = 0;
    int pointerP2 = 0;
    int pointerResult = start;

    while (!(pers1.length == pointerP1 || pers2.length == pointerP2)){ // loop
        while non of the pointer has reached the final element of it's array
        if (pers1[pointerP1].compareTo(pers2[pointerP2]) > 0 ){ // check
            if current (pointerP1) element from pers1 array is greater than the current element
            from pers2 array against > 0
                result[pointerResult] = pers2[pointerP2]; // put element
            from pers2 array in the result array
                pointerP2++; // move pointer of the
            2nd array to the next element

        } else {
            result[pointerResult] = pers1[pointerP1]; // put element
            from pers1 array in the result array
                pointerP1++; // move pointer of the
            1st array to the next element
        }
        pointerResult++; // update result
        pointer for next iteration
    }

    if (pers1.length == pointerP1) // check if array pers1 or pers2 has still
    remaining elements, which must be moved to result array
        System.arraycopy(pers2, pointerP2, result, pointerResult, (pers2.length -
        pointerP2)); // copy rest of array to result array // optional add rest of
        array via loop
    else
        System.arraycopy(pers1, pointerP1, result, pointerResult, (pers1.length -
        pointerP1)); // copy rest of array to result array
    }

}
```