Mergesort

Mergesort (1.5%)

Description

In this section, you are asked to implement FOUR functions

- `merge`
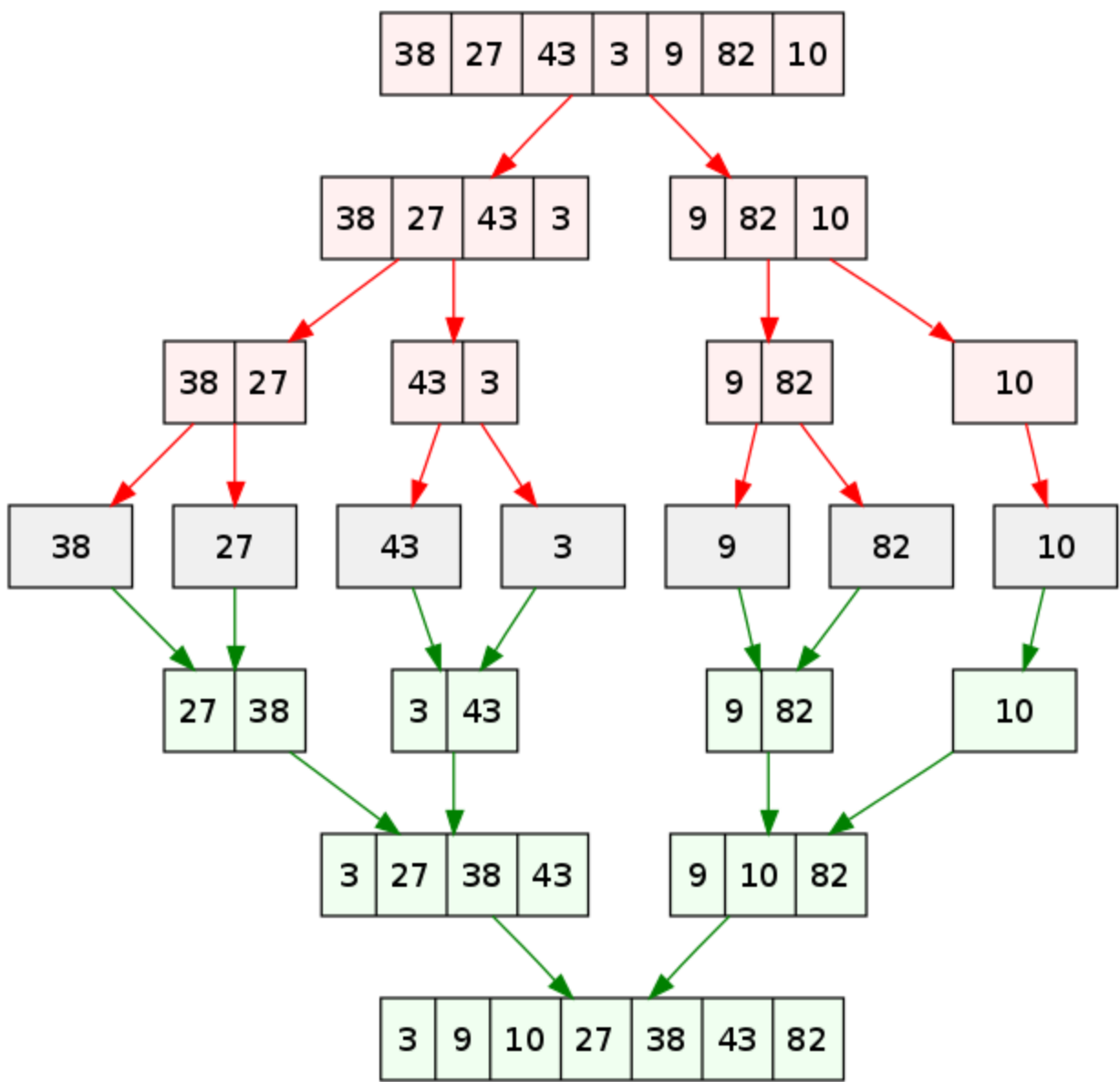
- `mergeSort`

- `mergeInplace`

- `mergeSortInplace`

This file only contains introductory information for implementing these four functions. More details regarding the spec of the functions and how to submit your work can be found in the section 6 of the auxiliary file.

PLEASE LISTEN CAREFULLY TO TAs!!!

Mergesort: Divide an array into two, sort each of them, and then merge two sorted arrays.

Given an array of numbers, sort them in non-decreasing order. You should implement mergesort. Other sorting algorithms are **not** allowed. You can refer to the following diagrams. You should not implement in place, that is, you should use extra memory.

Note: If you implement other sorting algorithms, we will find you, and we will penalize you. You will be penalized with negative scores. Don't even think about it!
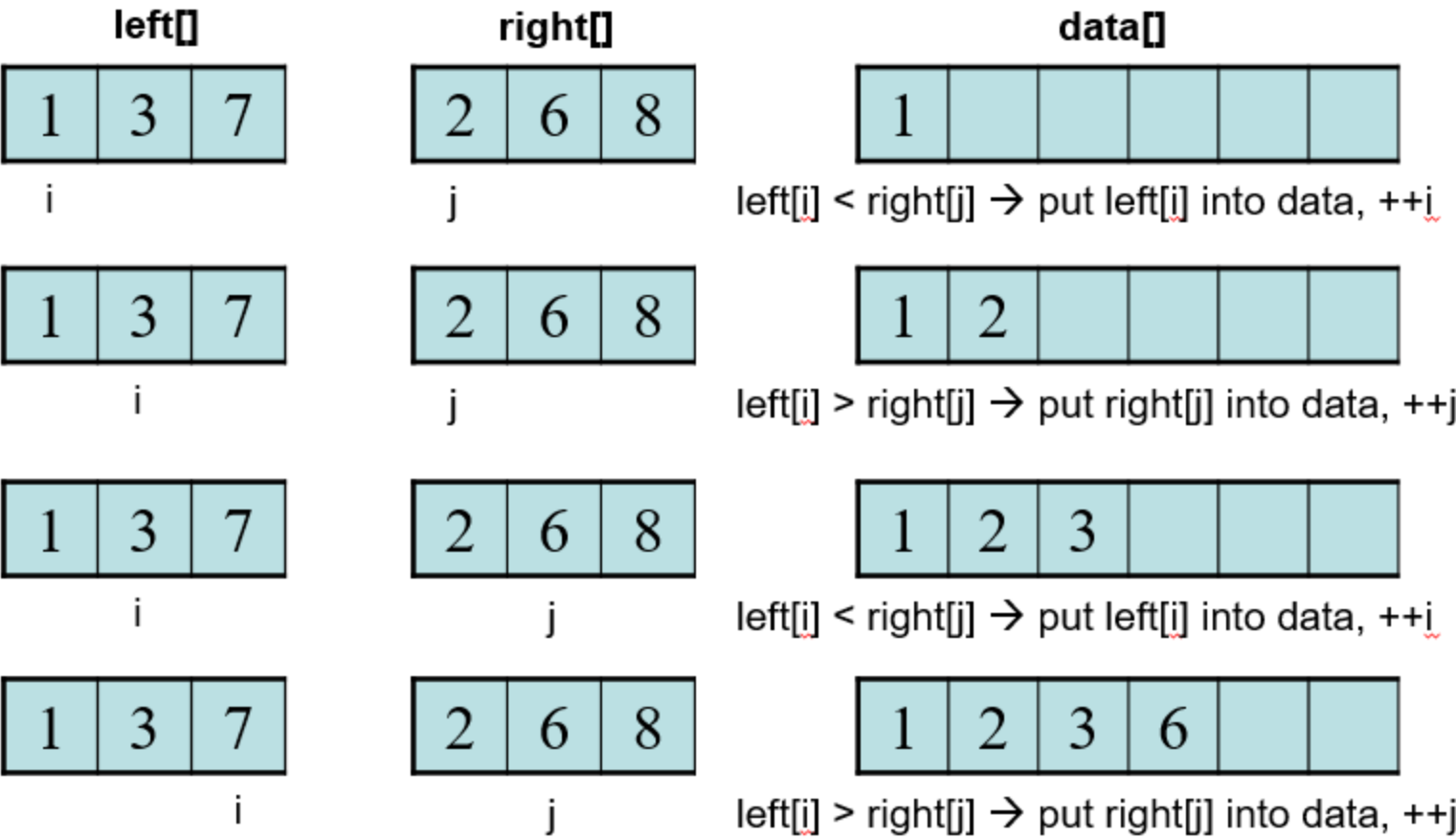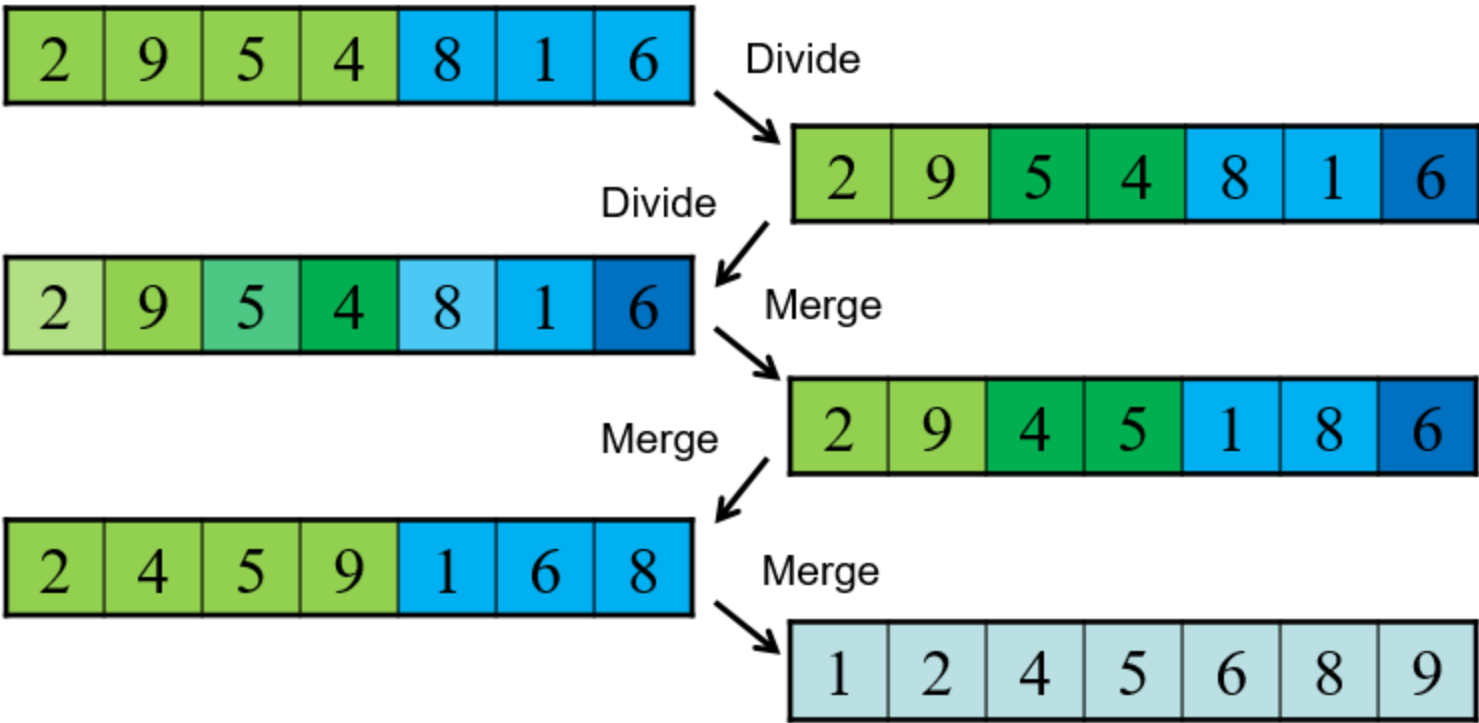
Normal version (0.5%)

merge

This function will merge two arrays into one array. Arguments are two arrays and their sizes.

```
{?} merge(?){
    /*
    your code here
    */
}
```

mergeSort

This function will Implement mergesort on the given array. Arguments are an array and its size.

```
void mergeSort(int* A, int size){
    /*
    your code here
    */
}
```

Inplace version (1%)

Inplace sorting

An in place sorting is a sorting method that takes $O(1)$ auxiliary space. It does not require extra memory. Using too much memory will cost our space, and we will not have enough space for other tasks in our computer. In order to save memory space, we should lower our memory usage. In the previous version, we use too much memory. Therefore, you should implement an in place version of mergesort now, that is, you **cannot** use extra memory.

Note: If you use extra memory space, we will find you, and we will penalize you. You will be penalized with negative scores. Don't even think about it!

mergeInplace

```
{?} mergeInplace(?){
    /*
    your code here
    */
}
```

mergeSortInplace

```
void mergeSortInplace(int* A, int size){
  /*
  your code here
  */
}
```