**2021.Algorithm – HW1**

**20172674**

**신동녘**

**1.**

In merge sort, there are 6-array access in total for the following reasons.

At first, we assume that there are N of data in the array, and we’ve got a temporary array in our function. This is how it works, at first, we should copy all of data from original array to temporary array while comparing them. If the comparing is end, we move the data existed in the temporary array to where it used to be. In this process, we access to the array for **“reading”** by copying in each steps(origin->temp, temp->origin), which means **2N** of access is executed. After copying data, we access to the array for **“writing”** by moving in each steps, this also gives **2N** of access. Finally, when comparing two components from divided arrays, **2N** of **“reading”** access is added.

For these reasons, we have **6N of access**, which include 4 reading access and 2 writing access, in top-down merge sorting.

**2.**

Ex) input -> arr = [10, 5, 4, 2, 6, 1, 7, 9, 3]

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

By increasing a length that we compare, cut the input array for the length we decide and put it into the merge function continuously. Precise explanation is in the comment written in the attached code file.