

On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.

서명: 강동인 학번: 21500002__

1. quiz1

Example: MERGESORTEXAMPLE

1. Improvement by reducing the number of merge() function call. Some hints for this problem are provided in the following pages.

```
// Improvement by reducing the number of merge() function call
void mergeSort(char *a, char *aux, int N, int lo, int hi) {
    if (hi <= lo) return;
    int mi = lo + (hi - lo) / 2;
    mergeSort(a, aux, N, lo, mi);
    mergeSort(a, aux, N, mi + 1, hi);
    if ( a[mi + 1] > a[mi] ) return; // already sorted
    merge(a, aux, lo, mi, hi);
}
```

M E R G E S O R T E X A M P L E
EM GR ES OR ET AX MP LE
EGMR EORS AETX ELMP
EEGMORRS AEELMPTX
AEEGLMMOPRRSTX

1. How many times did you spare merge() calls for "MERGESORTEXAMPLE" case?
 - Total number of merge() calls without your improvement: 15
 - The number of merge() calls spared with your improvement: 11
2. Identify those sets of char array groups that merge() call was unnecessary.

2. quiz2

```
//Example

// 5 1 7 3 2 8 6 4

// 15 37 28 46

// 1537 2468
```

```
//          12345678

int isSorted(int *a, int i, int j){return a[i] <= a[j];}

void merge(int *a, char *aux, int lo, int mi, int hi) {
    assert(isSorted(a, lo, mi)); // precondition: a[lo..mi] sorted
    assert(isSorted(a, mi+1, hi)); // precondition: a[mi+1..hi] sorted
    for (int k = lo; k <= hi; k++) aux[k] = a[k];
    .....
    assert(isSorted(a, lo, hi)); // postcondition: a[lo..hi] sorted
}
```

1. In the figure, which elements are compared in isSorted() at postcondition?

- First merge: 1, 5 | 3, 7 | 2, 8 | 4, 6
- Second merge: 1, 7 | 2, 8
- Third merge: 1, 8

2. Why isSorted() checks only two elements? Is this enough?

머지소트는 재귀를 사용합니다. 재귀는 기저단계와 리커시브 단계로 나뉘는데

기저 단계에서는 원소가 하나밖에 없으므로 비교할 필요가 없습니다.

두번째 단계에서는 비교 해야할 숫자가 2개가 됩니다. 올바르게 정렬되고

3번째 단계부터는 비교해야 할 숫자가 4개가 됩니다. 이미 전 단계에서 올바르게 정렬되었으니 3단계에서도 올바르게 정렬될 것입니다.

순차적으로 모든 단계를 마쳤을 때 각각의 스텝에서 정렬되어 있기 때문에 꼭 hi와 lo가 아닌 임의의 두 숫자를 비교하는 것만으로도 정렬이 되었는가를 판별하는데 충분합니다.