

```
void sort (int start, int end) {
```

```
    if (start == end) // 길이 1
```

```
        return;
```

```
    int mid = (start + end) / 2;
```

```
    sort (start, mid);
```

```
    sort (mid+1, end);
```

```
    merge (start, end);
```

```
}
```

[start, end]의 인덱스에 있는 값을 쪼개서
각각 합친후에 여기에 도착하면 절반씩은 이미

정렬되었으니 이들을 다시 합침.

→ 이미 정렬되어 자기 인덱스에
들어감.

```
void merge (int start, int end) {
```

```
    int mid = (start + end) / 2;
```

```
    int i = start, j = mid+1, k = 0;
```

→ 임시 배열 b를 위한 idx값.

```
    while (i <= mid && j <= end) {
```

```
        if (a[i] <= a[j]) b[k++] = a[i++];
```

```
        else b[k++] = a[j++];
```

```
    }
```

```
    while (i <= mid) b[k++] = a[i++]; // 한쪽만 남아있다면
```

```
    while (j <= end) b[k++] = a[j++]; // 반대쪽보다 큰 수들.
```

```
    for (int i = start; i <= end; ++i) {
```

```
        a[i] = b[i - start]; // 임시 배열 b에서 원래 배열 a의
```

인덱스 start부터 대입.

```
    }
```

```
}
```