```
// Sort A[0..N-1] into ascending order
void mergesort(int A[], int N) {
mergesort_help(A, 0, N-1);
}
// Sort A[lo..hi] into ascending order
void mergesort_help(int A[],int lo,int hi) {
if (lo < hi) {
int mid = (lo + hi) / 2;
mergesort_help(A, lo, mid);
mergesort_help(A, mid + 1, hi);
merge(A, lo, mid, hi);
}
}

// merge sequences A[lo..mid] & A[mid+1..hi],
// leaving merged result in A[lo..hi]
void merge(int A[], int lo, int mid, int hi){
int left = lo; int right = mid + 1;
int tempArray[MAX_SIZE]; //C++ notation . not valid Java
for (int i = 0; i <= hi-lo; ++i) {
assert (left <= mid || right <= hi);
assert (left <= mid+1 && right <= hi+1);
if (right == hi+1
|| (left <= mid) && (A[left] < A[right]))
tempArray[i] = A[left++];
else
tempArray[i] = A[right++];
}
for (i = 0; i <= hi-lo; ++i)
A[lo + i] = tempArray[i];
}
```