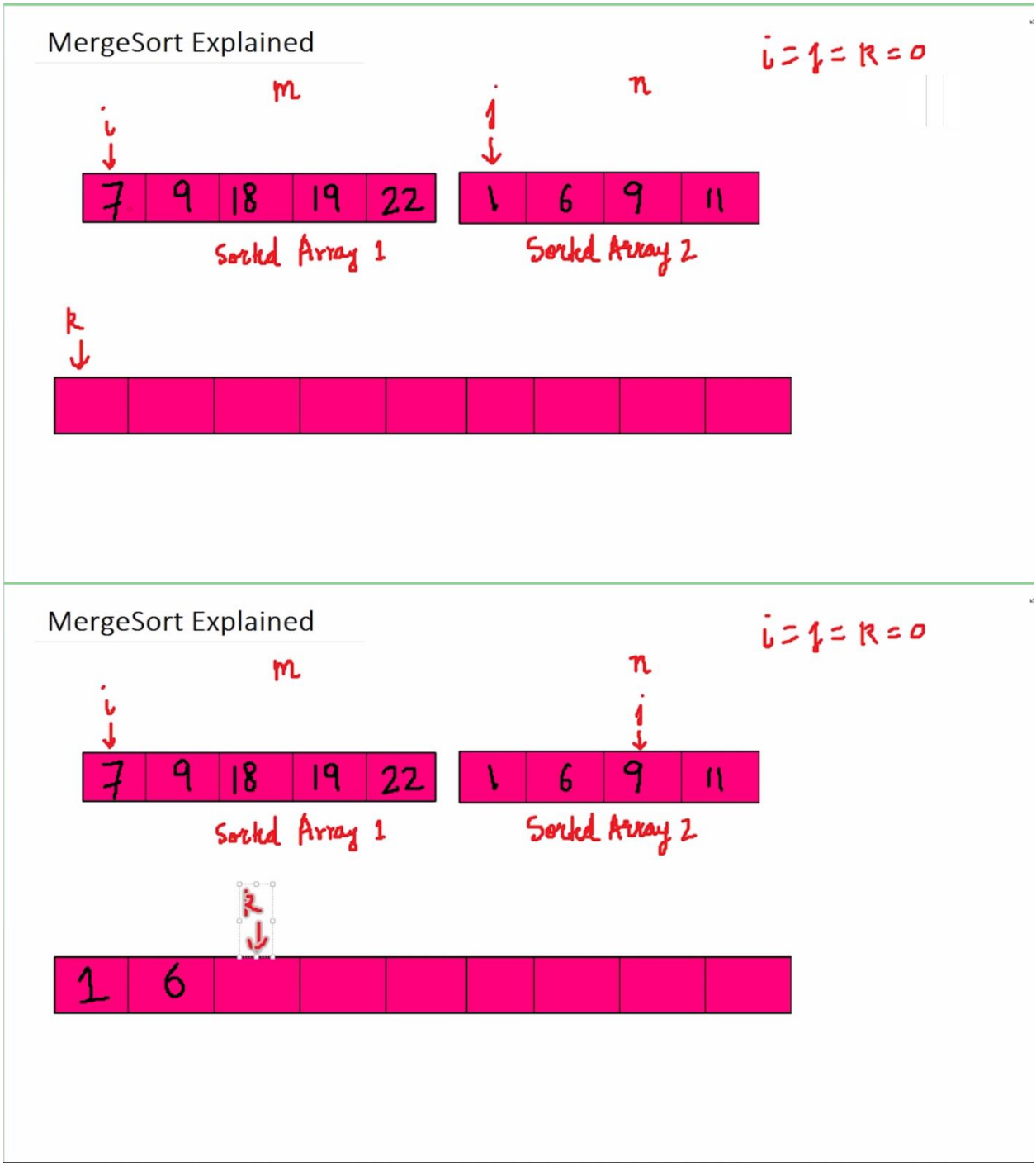
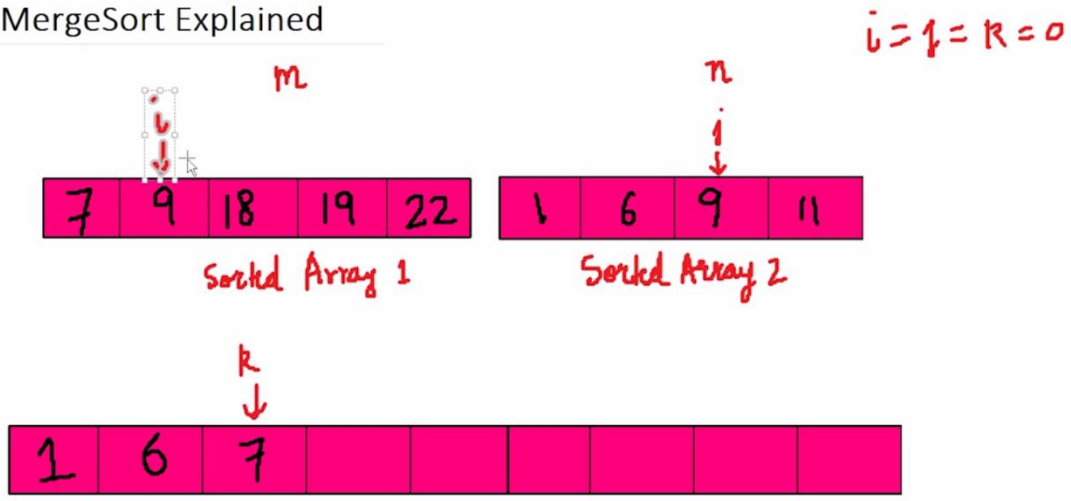
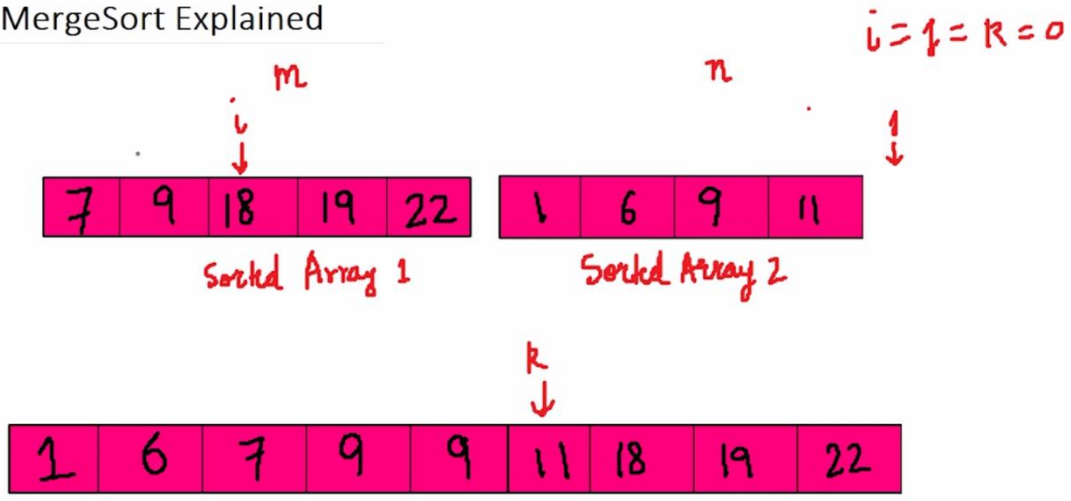
f

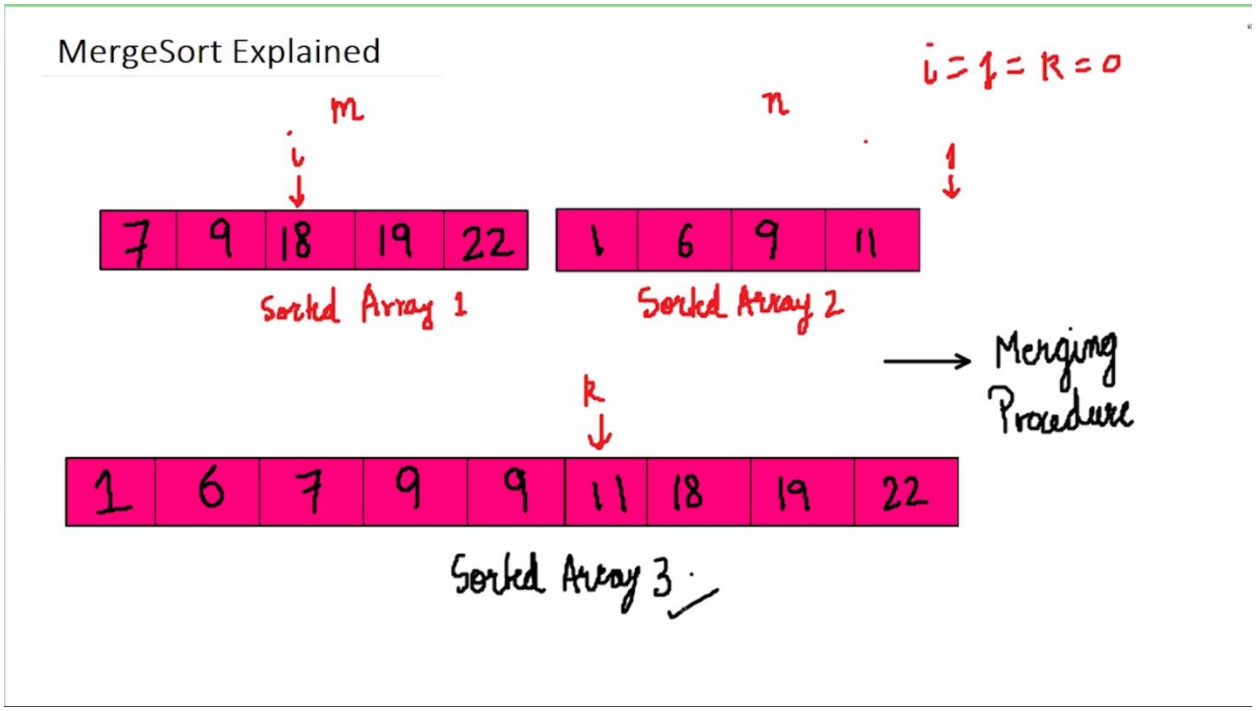

MergeSort Explained
i = j = k = 0
m
n
i
j
7 9 18 19 22
1 6 9 11
Sorted Array 1
Sorted Array 2
k
MergeSort Explained
i = j = k = 0
m
n
i
j
7 9 18 19 22
1 6 9 11
Sorted Array 1
Sorted Array 2
k
1 6

MergeSort Explained

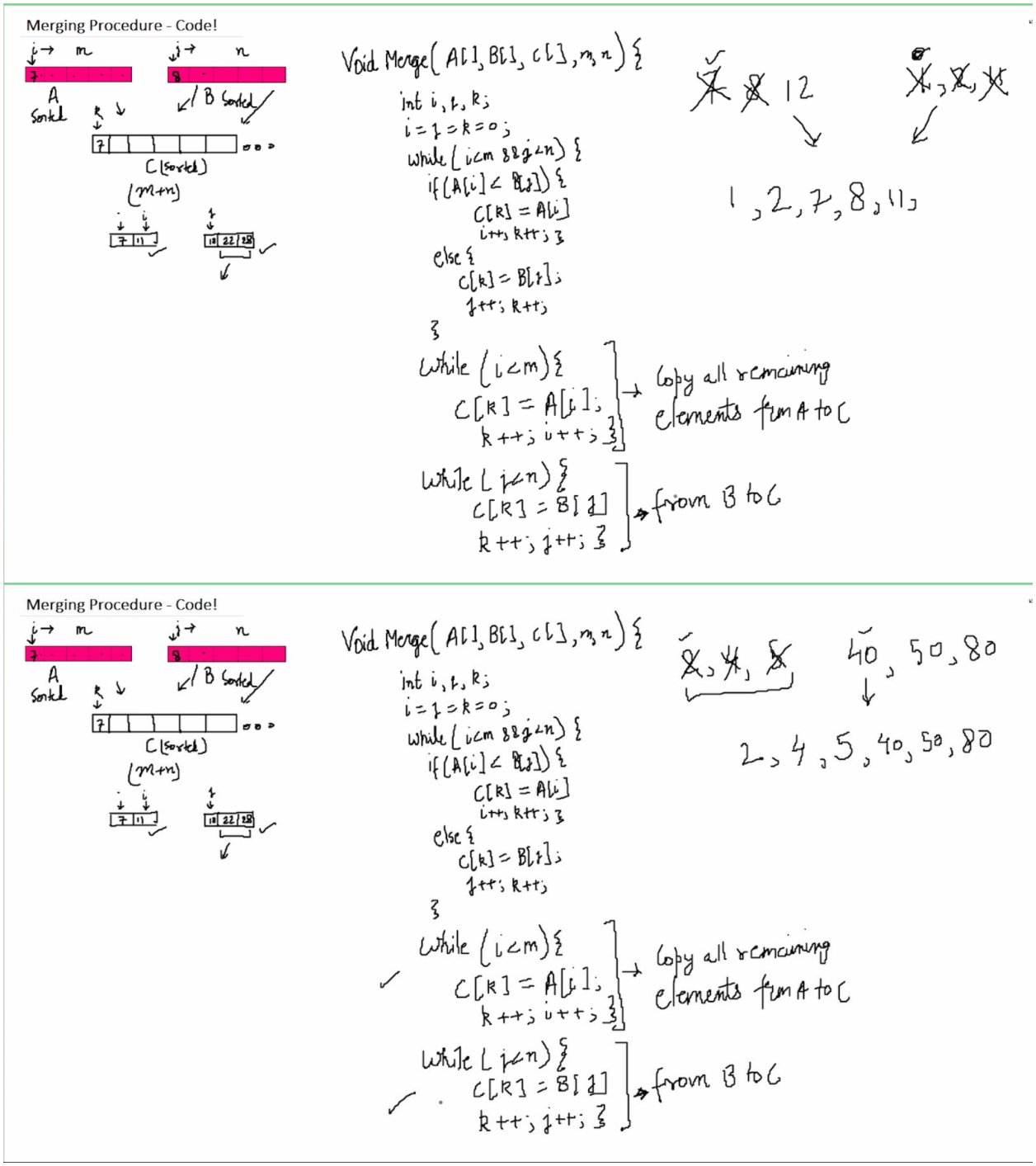


MergeSort Explained

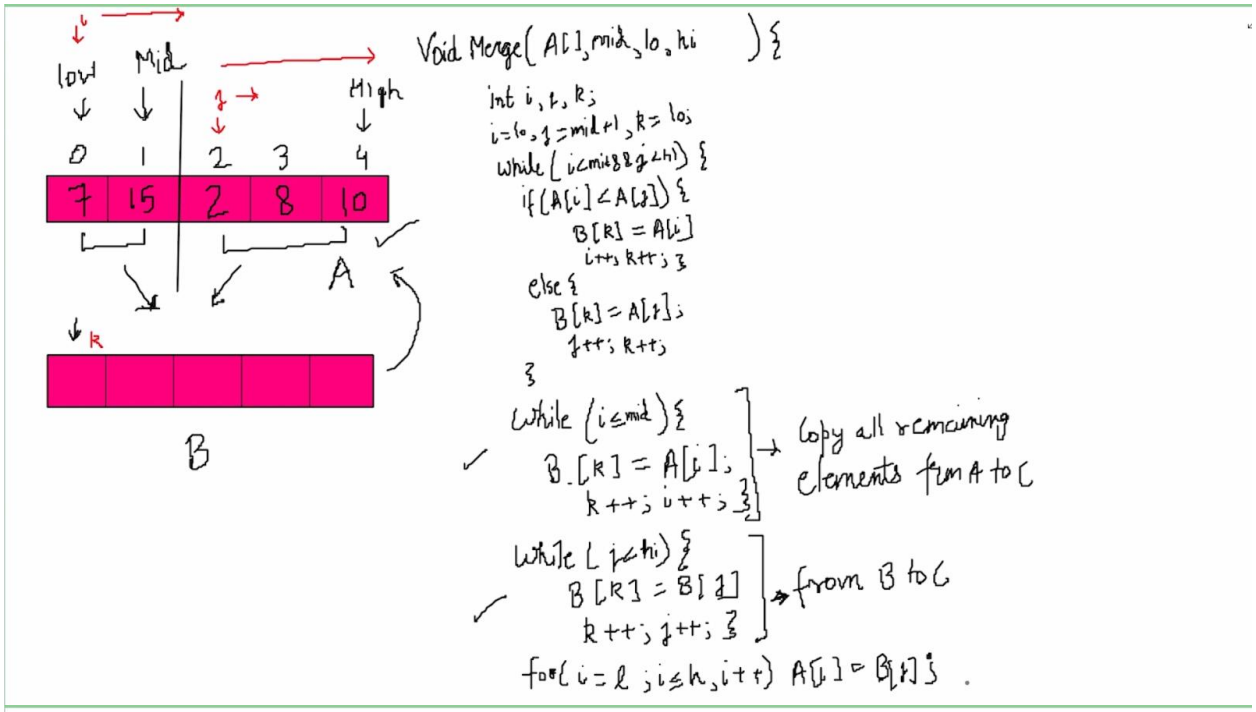


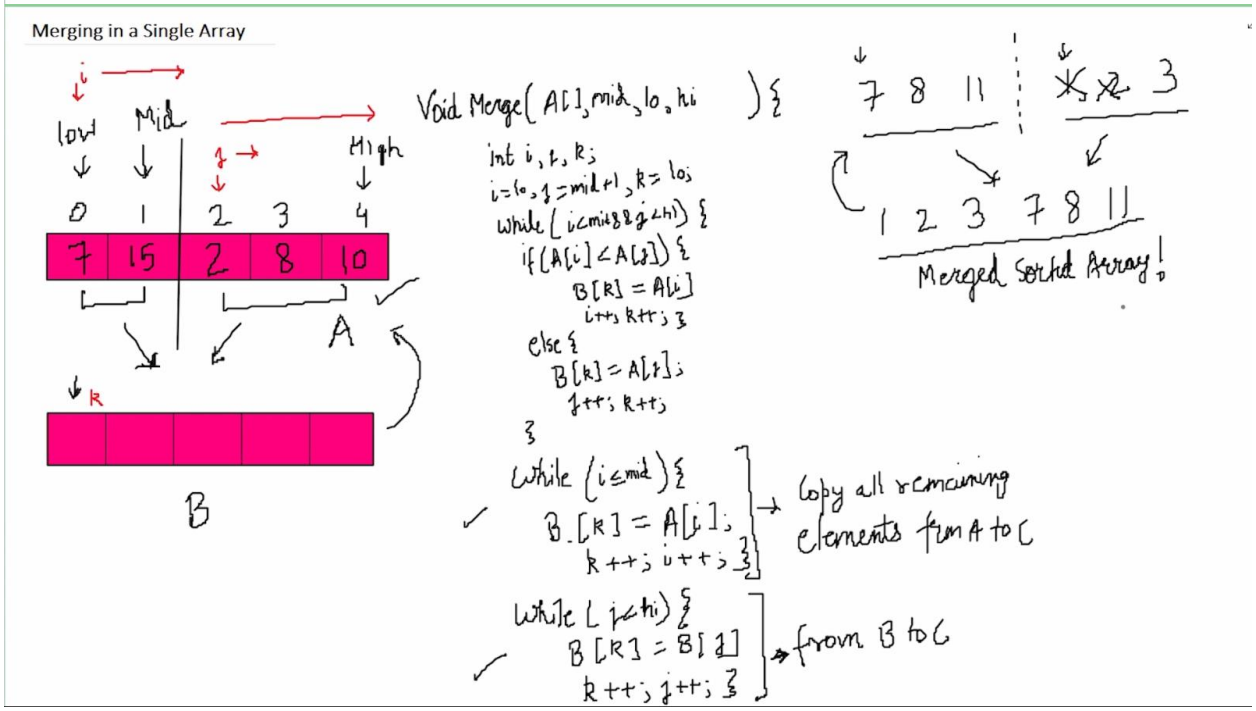

MergeSort Explained
i = j = k = 0
m
n
i
j
7 9 18 19 22
1 6 9 11
Sorted Array 1
Sorted Array 2
Merging Procedure
k
1 6 7 9 9 11 18 19 22
Sorted Array 3

Merge Procedure for code

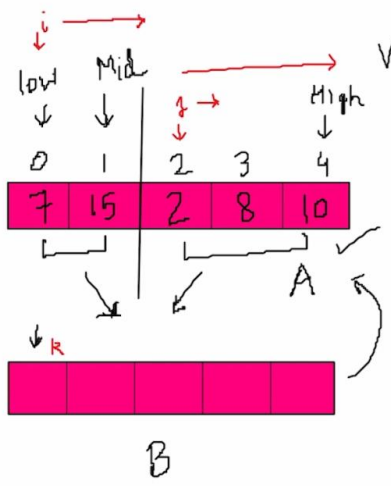


Merging in a single array





Merging in a Single Array




```
Void Merge( A[], mid, lo, hi ) {
  int i, j, k;
  i=lo, j=mid+1, k=lo;
  while (i<=mid && j<=hi) {
    if (A[i] < A[j]) {
      B[k] = A[i];
      i++; k++;
    }
    else {
      B[k] = A[j];
      j++; k++;
    }
  }
  while (i<=mid) {
    B[k] = A[i];
    k++; i++;
  }
  while (j<=hi) {
    B[k] = B[j];
    k++; j++;
  }
```

copy all remaining elements from A to C

from B to C

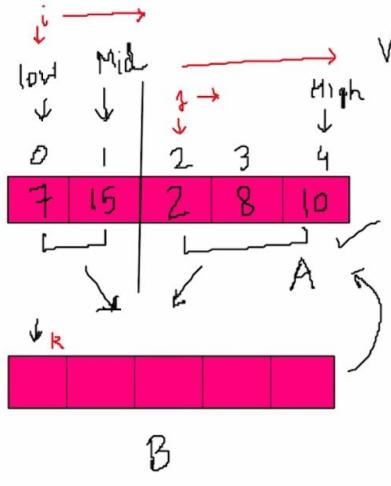
7 8 11 | 2 3

1 2 3 7 8 11

Merged Sorted Array!

11 7 ; 6 5

Merging in a Single Array




```
Void Merge( A[], mid, lo, hi ) {
  int i, j, k; int B[hi+1]
  i=lo, j=mid+1, k=lo;
  while (i<=mid && j<=hi) {
    if (A[i] < A[j]) {
      B[k] = A[i];
      i++; k++;
    }
    else {
      B[k] = A[j];
      j++; k++;
    }
  }
  while (i<=mid) {
    B[k] = A[i];
    k++; i++;
  }
  while (j<=hi) {
    B[k] = B[j];
    k++; j++;
  }
```

copy all remaining elements from A to C

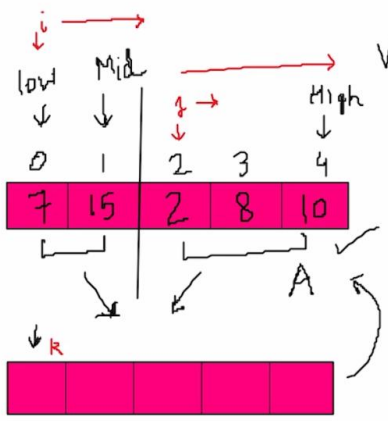
from B to C

7 8 11 | 2 3

1 2 3 7 8 11

Merged Sorted Array!

Merging in a Single Array

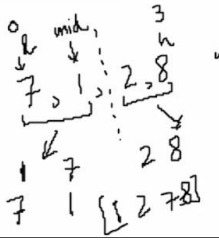



```
Void Merge( A[], mid, lo, hi ) {
  int i, j, k; int B[hi+1]
  i=lo, j=mid+1, k=lo;
  while ( i<=mid && j<=hi) {
    if (A[i] < A[j]) {
      B[k] = A[i]
      i++; k++; }
    else {
      B[k] = A[j];
      j++; k++;
  }
  while (i<=mid) {
    B[k] = A[i];
    k++; i++; }
```

→ copy all remaining elements from A to C

```
  while ( j<=hi) {
    B[k] = B[j]
    k++; j++; }
```

→ from B to C



Recursive Merge Sort

```
Void MS( int A[], int l, int h)
  if (l < h) {
    mid = (l+h)/2
    MS(A, l, mid);
    MS(A, mid+1, h);
    Merge(A, l, mid, h);
  }
```