

7. Merge sort

Theory-

- This sorting is the best of all sorting algorithms
- If list has one element then the list is sorted.
- We divide the list and sort the elements then we recombine or merge elements into a complete sorted list array.

Algorithm-

For array $A[]$ of n elements

$low = 0$ $high = n$ & $mid = (low + high) / 2$

Step 1:- If $low < high$ follow step 2 & 3

Step 2:- follow step 4 for $low = 0$ & $high = mid$ of $A[]$

Step 3:- follow step 4 for $low = mid + 1$ & $high = n$ of $A[]$

Step 4:- a) $mid \leftarrow (low + high) / 2$

b) $i \leftarrow low$ & $j \leftarrow mid + 1$

c) while $i \leq mid$ & $j \leq high$ follow (d) & (e)

d) if $A[i] < A[j]$ do $B[k] \leftarrow A[i]$

and $i \leftarrow i + 1$ & $k \leftarrow k + 1$

e) if $A[i] > A[j]$ do $B[k] \leftarrow A[j]$

and $j \leftarrow j + 1$ & $k \leftarrow k + 1$

f) while $i \leq mid$ do $B[k] \leftarrow A[i]$

$k \leftarrow k + 1$ & $i \leftarrow i + 1$

g) while $j \leq high$ $B[k] \leftarrow A[j]$

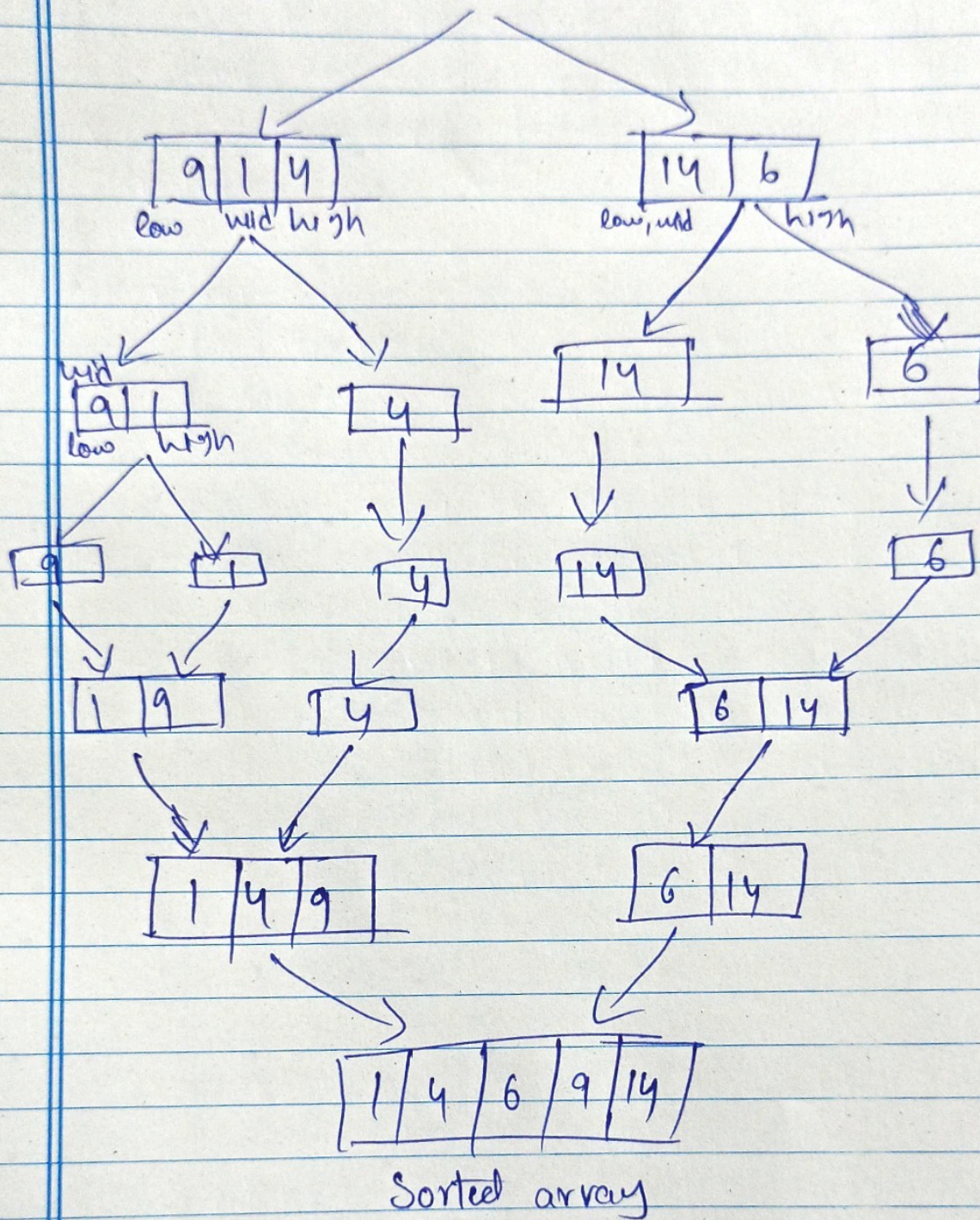
$k \leftarrow k + 1$ & $j \leftarrow j + 1$

h) Copy all $B[]$ array elements to $A[]$

process-

Array A:

9	1	4	14	6
---	---	---	----	---



Code -

```
#include <stdio.h>

int print_array(int A[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d", A[i]);
    printf("\n");
}

int merge(int A[], int low, int mid, int high) {
    int i, j, K, B[100];
    i = low;
    j = mid + 1;
    K = low;
    while (i <= mid & j <= high) {
        if (A[i] < A[j]) {
            B[K] = A[i];
            i++;
            K++;
        }
        if (A[j] <= A[i]) {
            B[K] = A[j];
            j++;
            K++;
        }
    }
    while (i <= mid) {
        B[K] = A[i];
        K++;
        i++;
    }
    while (j <= high) {
        B[K] = A[j];
        K++;
    }
}
```


Output-

Enter no. of elements: 9

Enter elements:

9
1
4
14
9
15
6

Before sorting: 9 1 4 14 4 15 6

After sorting: 1 4 4 6 9 14 15

```
    i++;  
    for (int i = low; i <= high; i++)  
        A[i] = B[i];
```

```
}  
int mergesort(int A[], int low, int high) {  
    int mid;  
    if (low < high) {  
        mid = (low + high) / 2;  
        mergesort(A, low, mid);  
        mergesort(A, mid + 1, high);  
        merge(A, low, mid, high);  
    }  
}
```

```
int main() {  
    int A[7], n;  
    printf("Enter no. of elements: ");  
    scanf("%d", &n);  
    printf("Enter elements: \n");  
    for (i = 0; i < n; i++)  
        scanf("%d", &A[i]);  
    printf("Before sorting:");  
    printArray(A, n);  
    return 0;  
}
```