Cyclomatic complexity for function *merge(int[] arr1, int[] arr2)*

1. int i = 0, j = 0, k = 0;

2. int[] arr3 = new int[2 \* n];

3. while (i < n && j < n) {

4. if (arr1[i] < arr2[j])

5. arr3[k] = arr1[i++];

6. else

7. arr3[k] = arr2[j++];

8. k++;

9. }

10. while (i < n)

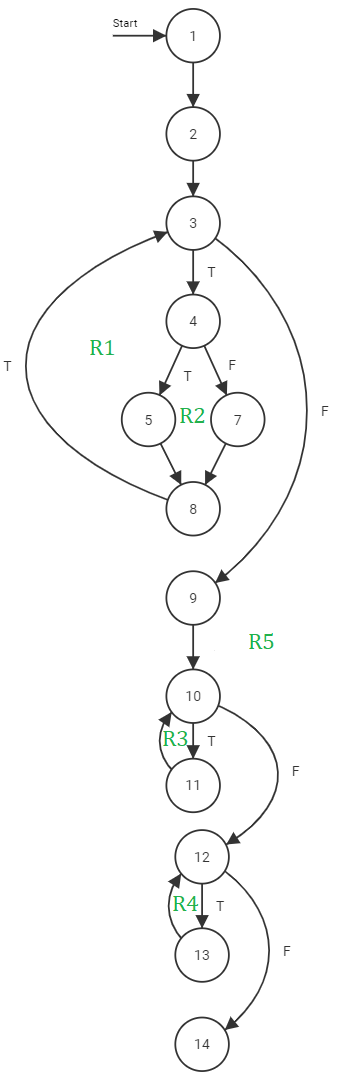
11. arr3[k++] = arr1[i++];

12. while (j < n)

13. arr3[k++] = arr2[j++];

14. return arr3;

**Control Flow Graph (CFG)**

****

**Cyclomatic complexity**

CC = edges – nodes + 2 = 16 – 13 + 2 = 5

CC = Number of regions from CFG = 5

Regions = {R1, R2, R3, R4, R5}

CC = P + 1 = 4 + 1 = 5 (P = predicate nodes)

P = {3, 4, 10, 12}

Independent paths:

Path 1: 1 – 2 – 3 – 9 – 10 – 12 – 14

Path 2: 1 – 2 – 3 – 4 – 5 – 8 – 3 – 9 – 10 – 12 – 14

Path 3: 1 – 2 – 3 – 4 – 7 – 8 – 3 – 9 – 10 – 12 – 14

Path 4: 1 – 2 – 3 – 4 – 7 – 8 – 3 – 9 – 10 – 11 – 10 – 12 – 14

Path 5: 1 – 2 – 3 – 4 – 7 – 8 – 3 – 9 – 10 – 12 – 13 – 12 – 14