MergeSort归并排序(A,b,e):

1. if b == e: return

2. m = (b + e) / 2 # 整数除法

3. MergeSort(A,b,m)

4. MergeSort(A,m+1,e)

# 把A[b..m] 和 A[m+1..e] 归并回 A[b..e]

5. for i = b,...,e: B[i] = A[i]

6. c = b

7. d = m+1

8. for i = b,...,e:

9. if d > e or (c <= m and B[c] < B[d]):

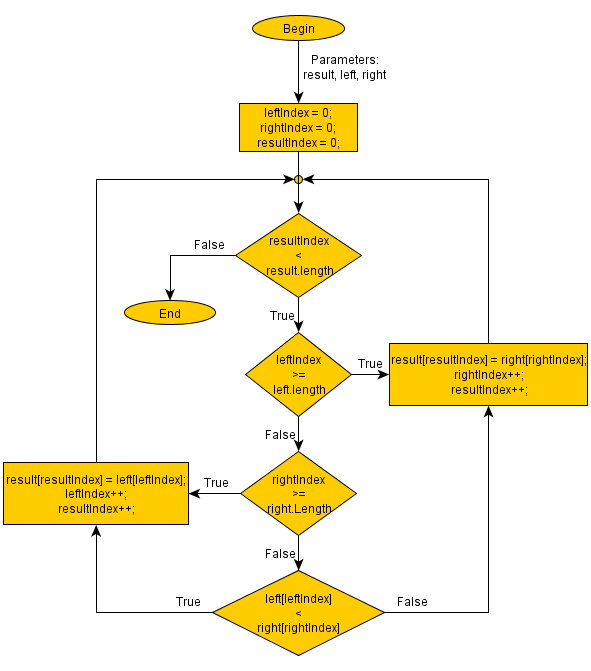
10. A[i] = B[c]

11. c = c + 1

else: # d <= e and (c > m or B[c] >= B[d])

12. A[i] = B[d]

13. d = d + 1



**Precondition（强条件）?**

b, e (- N, 0 <= b <= e < len(A))

elements of A[b..e] comparable with each other

A[b..e]中的元素可以被互相比较

**WP(predicate transformer)?**

merge sort(归并排序)

**Postcondition(弱谓语)?**

A[b..e] contains same elements as before, but sorted in

non-decreasing order (A[b] <= ... <= A[e])

A[b..e] 包含着与原来相同的元素，并且这些元素都被分类成为了递增的排序。

**Proof of correctness正确性证明:**

By induction on size n = e+1-b, prove (precondition and execution) implies (termination and postcondition), for all inputs of size n. Inductive structure of proof will follow recursive structure of algorithm.

通过使用归纳法，证明条件和程序运行的结果能够符合谓语的结果并且终止程序。对于所有的输 入n，归纳的证明结构将会遵循递归算法的结构。

**Base case基本条件**:

Suppose (A,b,e) is input of size n = e-b+1 = 1 that satisfies precondition. Then, e = b so algorithm terminates and returns A unchanged (on line 1), which satisfies postcondition.

**Ind. Hyp.递归的假设条件**:

Suppose n > 1 and, for 1 <= k < n, for all inputs of size k that satisfy precondition, algorithm terminates and postcondition holds after execution. (termination stands for total correctness)

**Ind. Step递归的假设证明**:

1. Suppose (A,b,c) is input of size n = e-b+1 > 1 that satisfies precondition, and consider call MergeSort(A,b,e).

2. Test on line 1 fails (because e-b+1 > 1 iff e > b) and algorithm

executes line 2.

3. Since b <= floor((b+e)/2) < e, IH implies that MergeSort(A,b,m)

terminates and A'[b..m] contains same elements as A[b..m] sorted in

non-decreasing order.

4. For the same reason, MergeSort(A,m+1,e) terminates and A'[m+1..e] contains same elements as A[m+1..e] sorted in non-decreasing order. (Notation: to compare elements of A at various points during execution, use A' to refer to order after execution of recursive calls.)

5. Line 5 copies A'[b..e] into B[b..e] (exercise: prove this).

6. Lines 6-13 merge B[b..m] and B[m+1..e] into A[b..e], which satisfies postcondition -- this requires formal proof, but we lack tools to carry this out for now...