---

**Algorithm 1 MergeSort**$(A, left, right)$

---

**Input:** 数组$A[1...n]$，数组下标$left, right$
**Output:** 递归数组$A[left..right]$

1: **if** $left \geq right$ **then**
2:      **return** $A[left..right]$
3: **end if**
4: $mid \leftarrow \lfloor \frac{left+right}{2} \rfloor$
5: MergeSort$(A, left, mid)$.
6: MergeSort$(A, mid + 1, right)$.
7: Merge$(A, left, mid, right)$.
8: **return** $A[left..right]$

---

**Algorithm 2 Merge**$(A, left, mid, right)$

---

**Input:** 数组$A[1...n]$，数组下标$left, mid, right$
**Output:** 递增数组$A[left..right]$

1: $A'[left..right] \leftarrow A[left..right]$
2: $i \leftarrow left, j \leftarrow mid + 1, k \leftarrow 0$
3: **while** $i \geq mid$ $and$ $j \geq right$ **do**
4:      **if** $A'[i] \geq A'[j]$ **then**
5:          $A[left + k] \leftarrow A'[i]$
6:          $k \leftarrow k + 1, i \leftarrow i + 1$
7:      **else**
8:          $A[left + k] \leftarrow A'[j]$
9:          $k \leftarrow k + 1, j \leftarrow j + 1$
10:      **end if**
11: **end while**
12: **if** $i \geq mid$ **then**
13:      $A[left + k..right] \leftarrow A'[i..mid]$
14: **else**
15:      $A[left + k..right] \leftarrow A'[j..right]$
16: **end if**
17: **return** $A[left..right]$

---

**Algorithm 3** 蛮力枚举法

**Input:** 数组$X[1...n]$
**Output:** 最大子数组之和$S_{max}$

1: $S_{max} \leftarrow -\infty$
2: **for** $l \leftarrow 1$ *to* $n$ **do**
3:      **for** $r \leftarrow l$ *to* $n$ **do**
4:          $S(l,r) \leftarrow 0$
5:          **for** $i \leftarrow l$ *to* $r$ **do**
6:              $S(l,r) \leftarrow S(l,r) + X[i]$
7:          **end for**
8:          $S_{max} \leftarrow \max\{S_{max}, S(l,r)\}$
9:      **end for**
10: **end for**
11: **return** $S_{max}$

---

**Algorithm 4** 优化枚举法

**Input:** 数组$X[1...n]$
**Output:** 最大子数组之和$S_{max}$

1: $S_{max} \leftarrow -\infty$
2: **for** $l \leftarrow 1$ *to* $n$ **do**
3:      $S(l,r) \leftarrow 0$
4:      **for** $r \leftarrow l$ *to* $n$ **do**
5:          $S \leftarrow S + X[r]$
6:          $S_{max} \leftarrow \max\{S_{max}, S\}$
7:      **end for**
8: **end for**
9: **return** $S_{max}$

**Algorithm 5 CrossingSubArray(X,low,mid,high)**

**Input:** 数组$X[1...n]$，数组下标$low, mid, high$
**Output:** 跨越中点的最大子数组之和$S_3$
  1: $S_{left} \leftarrow -\infty$
  2: $Sum \leftarrow 0$
  3: **for** $l \leftarrow mid$ *downto low* **do**
  4:     $Sum \leftarrow Sum + X[l]$
  5:     $S_{left} \leftarrow \max\{S_{left}, Sum\}$
  6: **end for**
  7: $S_{right} \leftarrow -\infty$
  8: $Sum \leftarrow 0$
  9: **for** $r \leftarrow mid + 1$ *to high* **do**
 10:     $Sum \leftarrow Sum + X[r]$
 11:     $S_{right} \leftarrow \max\{S_{right}, Sum\}$
 12: **end for**
 13: $S_3 \leftarrow S_{left} + S_{right}$
 14: **return** $S_3$

---

**Algorithm 6 MaxSubArray(X,low,high)**

**Input:** 数组$X[1...n]$，数组下标$low, mid, high$
**Output:** 最大子数组之和$S_{max}$
  1: **if** $low = high$ **then**
  2:     **return** $X[low]$
  3: **else**
  4:     $mid \leftarrow \lfloor \frac{low+high}{2} \rfloor$
  5:     $S_1 \leftarrow$ MaxSubArray(X,low,mid)
  6:     $S_2 \leftarrow$ MaxSubArray(X,mid+1,high)
  7:     $S_3 \leftarrow$ CrossingSubArray(X,low,mid,high)
  8:     $S_{max} \leftarrow \max\{S_1, S_2, S_3\}$
  9:     **return** $S_{max}$
 10: **end if**