Custom merge sort $\succeq$

```
Merge(A[1..n]: array of ints, l,m,r: indices in A; B[1..n]: array of ints,
      ⊕: двуместна релация)
@1  j <- l
@2  k <- m
@3  for i = l to r-1 do
@4      if (k=r) or (k<r & j<m & A[j] ⊕ A[k]) then
@5          B[i] <- A[j]
@6          j <- j+1
@7      else
@8          B[i] <- A[k]
@9          k <- k+1
@10     end if
@11 end for
```

Твърдение Ако $A[1..n]$ е масив от числа, $1 \le l \le m \le r \le n+1$ и $A[l..m-1]$ е сортиран във възходящ ред (↑) и $A[m...r-1]$ също е сортиран във възходящ ред. и още масивите A и B не споделят обща памет, то Merge(A, l, m, r, B) модифицира $B[l..r-1]$ до $B'[l..r-1]$, където $B'[l..r-1]$ е сортиран вариант на $A[l..r-1]$
При това A и $B[1..l-1]$ и $B[r...n]$ не се променят.

<u>Доказателство:</u> // 1) A не се променя, защото няма обща памет с B
2) Променя се само B и при това <u>само</u> $B[i]$ за $l \le i \le r-1$.
Следователно $B[1..l-1]$ и $B[r..n]$ не се променят

Нека j_s, k_s, i_s, B_s са състоянията на j, k, i, B непосредствено преди s-тото изпълнение на ред @3

Ред @3 се изпълнява $(r-1)-l+1 = r-l$ пъти

Дефинирам

↳ индукцията върви един ход напред от влизането в тялото на цикъла

$I(s) \overset{def}{\longleftrightarrow}$ 1) $B_s[l...i_s-1]$ е сортиран ↑ вариант на $A[l..j_s-1] \circ A[m...k_s-1]$ спрямо релацията ⊕

2) $j_s \le m$ 3) $k_s \le r$

4) $(j_s < m$ & $i_s > \ell \to B_s[i_s - 1] \oplus A[j_s])$
↳ поне едно завъртане на тялото на цикъла

5) $(k_s < r$ & $i_s > \ell \to B_s[i_s - 1] \oplus A[k_s])$

С индукция по s ще покажа, че $I(s)$ е истина за $1 \leq s \leq r - \ell$.

База: При $s=1$ 2) $j_1 = \ell \leq m$ (инициализирано на ред @1)

3) $k_1 = m \leq r$ (инициализирано на ред @1)

1) $B[\ell_1 \ldots \ell_1 - 1] = \{\}$ - сортиран ↑

4,5) предпоставката на импликацията е лъжа, цялата импликация е истина.

Нека $I(s)$ е изпълнено за някое непоследно достигане на ред @3.

Ще покажа, че $I(s+1)$ е истина.

→ Изпълнява се за s-ти път тялото на цикъла.

Тъй като при всяко изпълнение на тялото на цикъла се увеличава точно една от променливите j_s (на ред @6) или k_s (на ред @9), то общо $(k_s - m) + (j_s - \ell) = i_s - \ell$

Следователно ако $k_s = r$, то $r - m + j_s - \cancel{\ell} = i_s - \cancel{\ell}$

$r - i_s = m - j_s$, но $r > i_s$, защото се изпълнява тялото на for

следователно $j_s < m$

I сл.// if-а на ред @4 е истина $i_{s+1} = i_s + 1$ $j_{s+1} = j_s + 1$ $k_{s+1} = k_s \overset{2)}{\leq} r$

Тогава $B_{s+1}[i_s] = A[j_s]$ и от И.Х. 4) $B_s[i_s - 1] \oplus A[j_s]$ след. $B_s[i_s - 1] \oplus B_{s+1}[i_s]$

1) $B_s[\ell \ldots i_s - 1]$ е сортиран ↑ след. $B_{s+1}[\ell \ldots i_{s+1} - 1] = B_{s+1}[\ell \ldots i_s] = B_s[\ell \ldots i_s - 1] \circ B_{s+1}[i_s]$

след. $B_{s+1}[\ell \ldots i_{s+1} - 1]$ е сортиран ↑ и тъй като $j_{s+1} = j_s + 1$, то $B_{s+1}[\ell \ldots i_{s+1} - 1]$ е

сортиран ↑ вариант на $A[\ell \ldots j_{s+1} - 1] \circ A[m \ldots k_{s+1} - 1] = A[\ell \ldots j_s] \circ A[m \ldots k_s - 1]$ 1)

имаме, че $j_s < m$ след. $j_s + 1 \overset{3)}{\leq} m$.

→ масивът $A[\ell \ldots m-1]$ сортиран ↑

Ако $j_{s+1} < m$, то $B_{s+1}[i_{s+1} - 1] = B_{s+1}[i_s] = A[j_s] \oplus A[j_s + 1] = A[j_{s+1}]$ 4)

Ако $\begin{cases} j_{s+1} < m \\ k_{s+1} < r \end{cases}$, то $B_{s+1}[i_{s+1} - 1] = A[j_s] \oplus B[k_s]$ (усл. на if-а) 5)

II сл.// Аналогично

$$\boxed{Time_{Merge}(A, \ell, m, r, B) \in \Theta(r - \ell + 1)}$$

MergeSort (A[1…n]: array of integers; ℓ, r: indices in A,
B[1…n]: array of integers; $\oplus$: двуместна релация)

@1 $m \leftarrow \lfloor \frac{l+r}{2} \rfloor$
@2 if $r - l \le 1$ then return
@3 MergeSort(A, l, m, B)   // A[l..m-1]
@4 MergeSort (A, m+1, r, B)   // A[m...r-1]
@5 Merge(A, l, m, r, B)
@6 Copy (B, l, r, m)

Copy (B[1..n] : array of integers; l,r: indices; A[1..n]: array of integers)
@1 for i=l to r-1 do
@2 ⁝ A[i] <- B[i]
@3 done

Твърдение MergeSort (A, l, r, B) завършва за всеки $l \le r$ и ако A и B не споделят обща памет, MergeSort (A,l,r,B) сортира A[l..r-1], без да променя останалите елем. на A

Доказателство:// С инд. r-l

1) $r-l \le 1$. Тогава MergeSort завършва на @1. $A[l..r-1] \in \{A[l], \varepsilon\}$ и съответно е сортиран ↑.

2) Нека $r-l>1$

Тогава $m = \left\lfloor \frac{l+r}{2} \right\rfloor \le \frac{l+r}{2} < \frac{2r}{2} = r$

Следователно $m-l < r-l$

$$m = \left\lfloor \frac{r+l}{2} \right\rfloor \ge \frac{l+r-1}{2} > \frac{l+\overbrace{l+1}^{\text{вместо } r}}{2} = l$$

Следователно $r-m < r-l$

От. ИХ за m-l, MergeSort (A, l, m, B) сортира A[l...m-1] без да променя останалите елементи. Сега от свойството на Merge, след ред @5, B[l..r-1] е сортиран вариант на A[l..r-1] без да променя останала-та част от A. Следователно след @6 A[l..r-1] = B[l..r-1] ще сортиран вариант на входния A[l..r-1] без да промени останалите елем. на A.

Твърдение: Времевата сложност на MergeSort(A, l, r, B) е $\boxed{O\left((r-l+1).\log_2(r-l+1)\right)}$

Доказателство: Нека Time(m) е монотонно растяща функция по m

$$Time(m) = \max\{ Time_{MergeSort}(A, l, r, B) \mid r-l \le m \}$$

$$\boxed{\begin{aligned} T(2^{s+1}) &= 2.T(2^s) + c.s \\ T(2^s) &= 2^s(\alpha.s+\beta) = 2^s(\alpha.\log_2 2^s + \beta) \end{aligned}}$$

$Time(2^s) > Time(m)$ за $m \le 2^s$

Сега ако $r-l \le 2^s$, тогава $\boxed{\frac{r+l}{2} \le 2^{s-1}+l}$

$r-l \le 2^s \;/+2l$

$r+l \le 2^s + 2l \;/:2$

$\left\lfloor \frac{r+l}{2} \right\rfloor \le 2^{s-1}+l$

$\underbrace{\left\lfloor \frac{r+l}{2} \right\rfloor}_{m} - l \le 2^{s-1}$

Ако $r-l \le 2^s$, то $\left\lfloor \frac{r+l}{2} \right\rfloor = l + 2^{s-1}$ и

$r - \underbrace{\left\lfloor \frac{r+l}{2} \right\rfloor}_{m} = 2^{s-1}$

В противен случай $r-l \le 2^s-1$ и отново $r - \left\lfloor \frac{r+l}{2} \right\rfloor \le 2^{s-1}$

Тогава $Time(2^s) \le \underbrace{Time(2^{s-1})}_{\text{рекурсивно извикване за } l,m} + \underbrace{Time(2^{s-1})}_{\text{рекурсивно извикване за } m,r} + c.2^s$ } време за Merge и Copy

$Time(2^s) \le 2.Time(2^{s-1}) + c.2^s \quad (c>0)$

Дефинираме си редица, която да мажорира горната и правим у-е вместо неравенство.

Нека a_s е редицата

$a_0 = Time(2^0) = Time(1)$

$a_s = 2.a_{s-1} + c.2^s$

Тогава по индукция:

1) $a_0 \geq Time(2^0)$

2) ако $a_s \geq Time(2^s)$, то $a_{s+1} = 2.a_s + c.2^{s+1} \geq 2.Time(2^s) + c.2^{s+1}$

$\geq Time(2^{s+1})$

Решението на $a_s = 2a_{s-1} + c.2^s$ има общ. вид:

$$a_s = 2^s(\alpha s + \beta)$$

След. $Time(2^s) \leq a_s = 2^s(\alpha.s + \beta)$. Нека $n \in \mathbb{N}$

Избираме s така, че $2^{s-1} < n \leq 2^s$

Тогава $Time(n) \leq Time(2^s) \leq 2^s(\alpha.s + \beta) = 2.2^{s-1}(\alpha(s-1) + \alpha + \beta) <$

$2n(\alpha.\log_2 n + \beta + \alpha)$ // $2^{s-1} < n \leq 2^s$ /. $\log_2$

$s-1 < \log_2 n \leq s$

След. $Time(n) \in O(n.\log_2 n)$

От тук $Time_{MergeSort}(A, l, r, B) \in O((r-l+1).\log_2(r-l+1))$