

Сортировка перемешиванием

1) Инвариант для сортировки перемешивания:

Массив A, размера N

$A[\text{left_idx}-1] \leq A[\text{left_idx}] \ \&\& \ A[\text{right_idx}+1] \geq A[\text{right_idx}]$

Доказательство:

1. До начала цикла $\text{left_idx} = 0$, $\text{right_idx} = N$

$A[N] \geq A[N] \ \&\& \ A[0] \leq A[0]$

2. В первом внутреннем цикле после первой итерации right_idx уменьшается на единицу.

Тогда $A[N] \geq A[N-1]$, что справедливо, так как сравнивая элементы и меняя их местами на место $A[N]$ встанет наибольший элемент массива.

Во втором внутреннем цикле после первой итерации left_idx увеличивается на единицу.

Тогда $A[0] \leq A[1]$, что тоже справедливо, так как сравнивая элементы справа налево и меняя их местами на место $A[0]$ встанет наименьший элемент.

3. По математической индукции наш инвариант будет справедливым в любой момент цикла, так как мы доказали правильность для нулевой итерации и для первой.

4. Цикл всегда отработает за конечное время, так как right_idx уменьшается и одновременно left_idx увеличивается, наступит такая ситуация когда left_idx станет больше чем right_idx и цикл прекратится, инвариант останется справедливым.

2) Анализ сложности алгоритма перемешиванием.

Cost	N	Строки кода
C1	$N/2$	<code>for (size_t left_idx = 0, right_idx = size - 1; left_idx < right_idx;)</code>
C2	$\sum_{i=2}^{N/2} t[j]$	<code>for (size_t idx = left_idx; idx < right_idx; ++idx)</code>
C3	$\sum_{i=2}^{N/2} t[j]-1$	<code>if (array[idx + 1] < array[idx])</code>
C4	$\sum_{i=2}^{N/2} t[j]-1$	<code>std::swap(array[idx], array[idx + 1]);</code>
C5	$(N/2)-1$	<code>--right_idx;</code>
C6	$\sum_{i=2}^{N/2} t[j]$	<code>for (size_t idx = right_idx; idx > left_idx; --idx)</code>
C7	$\sum_{i=2}^{N/2} t[j]-1$	<code>if (array[idx - 1] > array[idx])</code>
C8	$\sum_{i=2}^{N/2} t[j]-1$	<code>std::swap(array[idx - 1], array[idx]);</code>
C9	$(N/2)-1$	<code>++left_idx;</code>

$$T(n) = C1 * N/2 + \sum_{i=2}^{N/2} t[j] * (C2 + C6) + \sum_{i=2}^{N/2} t[j]-1 * (C3+C4 +C7+C8) + (N/2-1) * (C5+C9)$$