

A4. Значительные инверсии

Демченко Георгий Павлович , БПИ-235

Условие

1. DaC-алгоритм CINV (CountInversions)

- [CountInversions.cpp](#)

Понял поставленную задачу, как подсчет кол-ва перестановок элементов до состояние отсортированности, а не поиск количества инверсий в массиве.

Суть алгоритма - модифицированный MERGE SORT

1.1 Шаг Divide

- Разбиваем входной массив на 2 подмассива по (примерно - ± 1) половине элементов в каждом.

Шаг CONQUER & COMBINE

- Рекурсивно выполняем подсчёт кол-ва необходимых перестановок между полученными подмассивами во время их сортировки (модифицированный MERGE из MERGE SORT)
- **Объединяем 2 подмассива в один**

1.2 $T(n) = O(n \cdot \log_2(n))$

- **MergeAndCountInversions**

$T_{maci}(n) = O(n)$ - линейное слияние и подсчёт кол-ва необходимых перестановок.

- **CountInversions**

$$\Rightarrow T(n) = 2 \cdot T\left(\frac{n}{2}\right) + T_{maci}(n) = 2 \cdot T\left(\frac{n}{2}\right) + O(n)$$

Согласно мастер-теореме

$$\log_b(a) = \log_2(2) = 1 = k$$

$\Rightarrow T(n) = O(n \cdot \log_2(n))$ - **Соответствует** требуемой ассимптотической верхней границе временной сложности

1.3 Минимальное количество необходимых инверсий.

Данный алгоритм не возвращает минимальное количество необходимых инверсий до состояния отсортированности

Так как подсчет количества необходимых инверсий ведется рекурсивно между двумя последовательными подмассивами изначального массива во время их слияния на каждом уровне рекурсии, то мы не можем гарантировать, что некоторая перестановка с элементом, не входящим в эти подмассивы, не будет оптимальней нашей (в плане кол-ва затраченных перестановок в дальнейшем). Также, при поднятии по уровням рекурсии, во время слияния мы не можем оставить перестановки, которые были необходимы для получения текущих отсортированных подмассивов и обязаны их учитывать.

В связи с этими факторами мы не всегда будем получать минимальное количество необходимых перестановок.

2. DaC-алгоритм CSINV (CountSignificantInversions)

Понял поставленную задачу, как подсчет кол-ва значительных инверсий (всех таких пар элементов), а не кол-ва перестановок как в предыдущем случае

Изменена логика подсчета инверсий в MergeAndCountSignificantInversions

- [CountSignificantInversions.cpp](#)

2.1 $T_{new}(n) = O(n \cdot \log_2(n))$

- **MergeAndCountSignificantInversions**

$T_{macsi}(n) = O(n)$ - линейное слияние и подсчёт кол-ва значительных инверсий.

- **CountSignificantInversions**

$$\Rightarrow T(n) = 2 \cdot T\left(\frac{n}{2}\right) + T_{macsi}(n) = 2 \cdot T\left(\frac{n}{2}\right) + O(n)$$

Согласно мастер-теореме

$$\log_b(a) = \log_2(2) = 1 = k$$

$\Rightarrow T_{new}(n) = T(n) = O(n \cdot \log_2(n))$ - **Соответствует** требуемой ассимптотической верхней границе временной сложности