

**СЛОЖНОСТ НА АЛГОРИТМИЧНИ ЗАДАЧИ**  
**КОНТРОЛНО № 4 ПО ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ**  
**(СУ, ФМИ, СПЕЦИАЛНОСТ “КОМПЮТЪРНИ НАУКИ”, I ПОТОК)**

**7 МАЙ 2021 Г.**

Докажете, че алгоритмичната задача за сортиране чрез сравнения на масив  $A[1 \dots n]$ , чиито елементи са два по два различни (т.е. няма повторения), изисква време  $\Omega(n \log n)$  при най-лоши входни данни.

**Решение:** Твърдението се доказва с помощта на подходяща редукция: общият случай (масив, сред чиито елементи може да има равни) се свежда до разглеждания частен случай (масив, чиито елементи са два по два различни). Както е известно, общият случай изисква време  $\Omega(n \log n)$ .

```
Sort (A [1...n] : произволен масив) // общият случай
B [1...n] : масив от наредени двойки <value, index>
for k ← 1 to n
    B [k].value ← A [k]
    B [k].index ← k
SortDistinct (B [1...n]) // частният случай — няма повторения
for k ← 1 to n
    A [k] ← B [k].value
```

*Коректност* на редукцията: Първият цикъл прави елементите различни: всеки два елемента на масива  $B$  се различават по стойността на полето `index`. Следователно на алгоритъма `SortDistinct` се подават допустими входни данни.

Предполагаме, че има някаква наредба върху базовия тип на масива  $A$ ; иначе задачата няма смисъл. Обаче трябва сами да дефинираме наредба върху базовия тип на масива  $B$ . Подходяща за целта е лексикографската наредба: наредените двойки `<value, index>` се сравняват по полето `value` (като елементи на масива  $A$ ); ако стойностите на полетата `value` са равни, тогава се сравняват стойностите на полето `index` (като цели числа). Следователно сортирането на  $B$  е равносилно на устойчиво сортиране на  $A$ , т.е. редукцията е коректна.

*Бързина* на редукцията: Всеки от двата цикъла обхожда масивите, а всяко обхождане изисква линейно време. Тоест времевата сложност на редукцията е  $\Theta(n) = o(n \log n)$ , което е достатъчно бързо за целите на доказателството.

**Схема за точкуване:**

Контролното носи 4 точки: 1 т. за описание на редукцията на псевдокод; 1 т. за анализ на бързината ѝ; 2 т. за доказване на коректността ѝ (от които 1 т. за дефиниране на наредба на  $B$  и 1 т. за доказване, че `SortDistinct` се вика с допустими входни данни).