ДАА семинар 15

Зад Даден е масив $A[1..n]$, чиито елементи са сортирани. Да се намери долна граница за всеки алгоритъм, използващ функцията Compare(i,j), която връща $<, =$ или $>$ в зависимост от резултата от сравнението на $A[i]$ и $A[j]$.

Решение:

Ще покажем, че $n-1$ е долна граница за броя на сравненията (извикванията на Compare)

// Правване да си измислим максимално ефикасен алг., който да решава задачата. В случая най-бързо и лесно е линейно да обходим масива и да сравним съседните ел. Оттук ще получим $n-1$ сравнения

Нека AlgEUSort е алгоритъм, който решава задачата и който прави не повече от $n-2$ сравнения.

Да разгледаме случая с $n-2$ е достатъчно.

Ще използваме аргументация с противник:

Броят на съседните двойки елементи в $A[1..n]$ е $n-1$.
Щом са направени $n-2$ сравнения, то това означава, че има поне една двойка съседни елементи, които не са били сравнени.

Нека $A[i] = 2i$ за $i = 1..n$. Противникът отговаря с изхода от ф-ята Compare.

- Ако алг. отговори с „Да", т.е че елементите са уникални, то противникът ще го опровергае като за двойката $A[i]$ и $A[i+1]$, които не са били сравнени зададе стойност $A[i] = A[i+1] = 2i$.

По този начин отговорите му са консистентни – $A[1..n]$ е сортиран, но алгоритъмът е сгрешил.

– Ако алг. отговори с „Не", то противникът показва оригиналният $A[1...n]$

2 зад Покажете, че $2n-1$ е долна граница за броя на сравненията при сливането на $A[1..n]$ и $B[1..n]$ в един сортиран масив $C[1...2n]$.

Решение: Долна граница $2n-1$ ни подсказва, че всяка двойка съседи в $C[1..2n]$ е била сравнена.

Отново ще използваме аргументация с противник, възползвайки се от този факт.

Нека Alg X е алгоритъм, който слива двата масива, правейки $2n-2$ сравнения.

Нека $A[i] = 2i-1$ за $i = 1..n$ и
$B[i] = 2i$ за $i = 1..n$.

//Зад: Не е нужено да се посочват точно такива стойности. Така обаче е по-ясно, без да има загуба на общност.

Щом сравненията са $2n-2$, то има двойка съседни елементи, които не са били сравнени.

При този вход Alg X е върнал масива:

$C = [A[1], B[1], A[2] \dots A[n], B[n]]$

Възможни са 2 случая за съседи, които не са сравн.

— $A[i]$ не е бил сравнен с $B[i]$:

Противникът задава стойност $A[i] = 2i$, $B[i] = 2i - 1$
Така наредбата в масивите A и B не се променя, т.е
той не може да бъде уличен в лъжа, но
Alg X връща сейсия C, което е грешен резултат.

— $A[i]$ не е бил сравнен с $B[i-1]$: Аналогично

на горния случай: $A[i] = 2(i-1)$ $B[i-1] = 2i - 1$

Зад. 3 Дадено е множество от наредени двойки $X = \{(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\}$, такива че $a_i < b_i$ за $1 \leqslant i \leqslant n$. Известно е, че числата $a_1, \dots, a_n, b_1, \dots, b_n$ са две по две различни. За всеки две наредени двойки (a_i, b_i) и (a_j, b_j), където $i \neq j$, казваме, че са *независими*, ако е изпълнено $a_i < b_i < a_j < b_j$ или $a_j < b_j < a_i < b_i$. Ако е известно, че наредените двойки $(a_{i_1}, b_{i_1}), (a_{i_2}, b_{i_2}), \dots, (a_{i_k}, b_{i_k})$ са две по две независими, казваме, че масивът от тях

$$[(a_{i_1}, b_{i_1}), (a_{i_2}, b_{i_2}), \dots, (a_{i_k}, b_{i_k})]$$

е *растящ*, ако

$$a_{i_1} < b_{i_1} < a_{i_2} < b_{i_2} < \dots < a_{i_k} < b_{i_k}$$

Професор Интервалски твърди, че разполага с алгоритъм, който във време $O(n \lg \lg n)$ намира множество $Y \subseteq X$, такова че наредените двойки в Y са две по две независими и $|Y| \geqslant |Z|$ за всяко $Z \subseteq X$, такова че наредените двойки в Z са независими. Нещо повече, професорът твърди, че неговият алгоритъм връща Y в растящ масив. Опровергайте го: докажете, че такъв алгоритъм няма.

Решение: Interval scheduling от лекции