主題

日期

星期

Algorithm Analysis and design HW-4:

5.1 Name the two sets A and B

Suppose $A = \{a_1, a_2, \cdots, a_n\}$. $B = \{b_1, b_2, \cdots, b_n\}$

and $a_1 > a_2 > \cdots > a_n$, $b_1 > b_2 > \cdots > b_n$.

Lemma: Suppose the n^{th} smallest value is in A.

Then the n^{th} smallest value is a_k <=>

$b_{n-k} > a_k > b_{n-k+1}$

Proof: "=>" a_k is the n^{th} smallest value and there are

$(k-1)$ values in A larger than a_k

So, there are exactly $n-k-1$ values in B

larger than a_k, which means $b_{n-k} > a_k > b_{n-k+1}$

"<=" (The same as above)

Given k, Let $c_k = b_{n-k} - a_k$, $d_k = b_{n-k+1} - a_k$

Then we need to find k s.t. $c_k > 0$. $d_k < 0$.

Notice that c_k increases monotonically with k and

d_k decreases monotonically with k

So we can use binary search algorithm to find the correct index k

主題	日期
	星期

① If k exists, then a_k is the answer

② If there's no such k, then use the same way to find t s.t. $a_{n-t} > b_t > a_{n-t+1}$

And we'll get the correct answer b_t.

The time cost is $O(\log n)$. which lies the same as binary search.

5.2. We can achieve this by simply changing the sort and count algorithm like this:

Sort-and-Count(L)

If the list has one element then

there are no inversions

Else

Divide the list into two halves:

A contains the first $\lceil n/2 \rceil$ elements

B contains the remaining $n - \lceil n/2 \rceil$ elements

$(r_A, A) =$ Sort-and-Count(A)

$(r_B, B) =$ Sort-and-Count(B)

$(r, L) =$ Merge-and-Count(A, B)

$(r', L') =$ Merge-and-Count(A, B·2)

主題

日期

星期

Endif

Return $r = r_A + r_B + r'$ and the sorted list L