Re: On the complexity of determining whether n numbers are distinct

Source: http://coding.derkeiler.com/Archive/General/comp.theory/2006−09/msg00203.html

- *From*: "eKo1" <berndlosert@xxxxxxxxxxxx>
- *Date*: 25 Sep 2006 12:06:42 −0700

Googmeister wrote:

> eKo1 wrote:
>
>> There are two algorithms I can think of that determines whether n numbers are distinct:
>>
>> 1. Compare each number with every other number. If you find two that are the same, the numbers are not distinct.
>> 2. Sort the numbers. Loop through the sorted list and if you find that the ith and (i + 1)th entries in the list are the same, the numbers are not distinct.
>>
>> The number of comparisons in the first algorithm is proportional to n^2 and in the second to n lg n in the worst−case (viz. that all numbers are distinct).
>>
>> I want to prove that any algorithm A, in the worst−case, has to do at least a number of comparison s proportional to n lg n. I've read that this can be shown by reducing the problem to one of sorting but the proof is bogus because it takes A, changes it to A' (which does sorting) and then incorrectly states that both A and A' contain the same number of comparisons.
>
> Before proving it, you have to be precise about your model of computation, e.g., what operations are permitted. The paper "Lower bounds for algebraic computation trees" by Michael Ben−Or proves the n log n lower bound in the algebraic decision tree model of computation.

I didn't know that the lower−bound was dependent on the model of computation. The book where I got this problem from does mention decision trees and uses it proves that the lower−bound of comparison−based sorting is Big−Omega(n lg n). I will definitely be looking into this.

Note to all: I'm dealing with comparison−based algorithms in my first post.

.