

Complexiteit

Uitwerkingen Opgaven

Opgave 3e

We moeten bewijzen: *elk* algoritme voor het zoekprobleem dat is gebaseerd op het doen van sleutelvergelijkingen doet in de worst case ten minste n sleutelvergelijkingen. Zo'n algoritme werkt voor elk mogelijk invoerarray, en geeft daarop –uiteraard– het correcte antwoord.

We bewijzen de bewering via een bewijs uit het ongerijmde.

Dus: veronderstel dat de stelling niet waar is; dan is er dus een algoritme voor het zoekprobleem dat in de worst case hooguit $n - 1$ vergelijkingen doet, en dus doet dat algoritme voor *elke* invoer $n - 1$ vergelijkingen of minder. We laten zien dat dit tot een tegenspraak leidt door 2 invoeren te geven waarop het algoritme ten onrechte hetzelfde antwoord geeft.

Bekijk als invoer (noem die even invoer1) een ongesorteerd array A , bestaande uit n verschillende waarden, en een X die niet in het array zit. Het algoritme zal op deze invoer dus antwoord -1 geven (hetgeen betekent dat X niet in A zit). Aangezien het algoritme volgens de aanname hooguit $n - 1$ sleutelvergelijkingen doet, zal ten minste één array-element niet zijn bekeken; zeg dat dit $A[j]$ is. Bekijk nu een invoerarray B dat precies gelijk is aan invoer1, behalve dat nu op plek j wel X staat: $B[i] = A[i]$ voor alle $i \neq j$ en $B[j] = X$. Laat nu het algoritme los op B . Op deze invoer doet het algoritme precies hetzelfde als op invoer1. Immers, aangezien $A[j]$ nooit bekeken werd en alle andere $B[i]$'s gelijk zijn aan de $A[i]$'s en X hetzelfde is als voorheen, zijn de antwoorden op de gestelde vragen/sleutelvergelijkingen hetzelfde en worden ook dezelfde vragen/sleutelvergelijkingen gedaan als voor invoer1¹. Het algoritme zal dus ook op deze invoer antwoord -1 geven. Dat is in dit geval fout, omdat X wel degelijk in B voorkomt. Dit is in tegenspraak met de correctheid van ons algoritme. Ergo, de aanname was verkeerd, en dus geldt: elk algoritme voor het zoekprobleem doet in de worst case ten minste n sleutelvergelijkingen. QED.

Samengevat, als we aannemen dat er een algoritme bestaat dat altijd $\leq n - 1$ sleutelvergelijkingen doet, kunnen we twee verschillende invoeren construeren waarop het algoritme ten onrechte hetzelfde antwoord geeft. Aangezien dit in tegenspraak is met het feit dat het algoritme voor elke invoer correct werkt, was de aanname verkeerd. En klaar.

Opmerking: we hebben in het bewijs gebruikt dat we helemaal niets weten over het array. Je kunt alleen informatie over een arrayelement krijgen door dat element rechtstreeks te vergelijken met X . Als je een element niet vergeleken hebt weet je er ook niets van. Het bewijs gaat bijvoorbeeld niet op voor arrays waarvan we weten dat ze gesorteerd zijn, dus waarvan we de ordening weten. In dat geval hoeft je een array-element $A[\ell]$ niet direct met X vergeleken te hebben om te kunnen concluderen dat $A[\ell] \neq X$. Voor willekeurige arrays is dat wel het geval.

¹Stel het algoritme vraagt of $B[i] = X$; dat is net als bij invoer1 ($= A$) niet het geval; dus gaat het algoritme op B precies hetzelfde verder als op A ; dit gaat herhaald zo door. Het algoritme zal in het bijzonder dan ook $B[j]$ nooit bekijken