

Tentamen Algoritmen en Datastructuren

dinsdag 3 november 2009, 14 - 17 uur

Elk vraagstuk levert maximaal 18 punten op. Het tentamencijfer T is $(p/10) + 1$, waarbij p het totaal aantal behaalde punten is.

Met de zinsnede 'geef een algoritme' in een opgave wordt bedoeld:

beschrijf een algoritme in pseudocode, licht de werking ervan toe, beargumenteer de correctheid.

1. We gebruiken een hash-tabel $T[0..m-1]$ voor het opslaan van niet-negatieve gehele getallen. De hashfunctie h is gedefinieerd door $h(i) = i \bmod m$. We gebruiken *open addressing* met *linear probing* voor het vermijden van botsingen (*collisions*). Dat wil zeggen: als getal i in de tabel geplaatst moet worden en positie $h(i)$ is niet leeg, dan worden de posities $h(i+1), h(i+2), \dots, h(i+m-1)$ langsgelopen en wordt i geplaatst in de eerste lege positie. We nemen aan dat er altijd een lege positie in de tabel is.

Het algoritme voor het opzoeken van een getal in de hash-tabel is als volgt.

```
algoritme ZoekOp( $n, T$ )
  invoer getal  $n$ , hash-tabel  $T[0..m-1]$ 
  uitvoer de positie  $k$  met  $T[k] = n$  als die er is, anders de melding 'is er niet'
   $k \leftarrow n \bmod m$ 
  while  $T[k] \neq \text{NULL}$  do
    if  $T[k] = n$  then
      return  $k$ 
     $k \leftarrow (k + 1) \bmod m$ 
  return 'is er niet'
```

Geef efficiënte algoritmen voor het toevoegen en verwijderen van een getal in resp. uit de hash-tabel. Houd rekening met het ontstaan van 'gaten' bij het verwijderen van getallen.

2. Gegeven is een lijst S van n getallen uit het interval $[0, n^2 - 1]$. Geef een eenvoudig algoritme dat S sorteert in $\mathcal{O}(n)$ tijd.
3. Geef het algoritme $\text{DFS}(G, v)$ voor *depth-first search* vanuit knoop (*node*) v in een ongerichte samenhangende graaf G . DFS kent aan elke kant (*edge*) een label *discovery* of *back toe*: de *discovery edges* vormen een opspannende boom. Laat zien dat, bij een geschikt gekozen representatie van G , de complexiteit van het algoritme $\mathcal{O}(n + m)$ is, met m het aantal kanten en n het aantal knopen van G .
4. Beschrijf de compacte representatie van de gecomprimeerde suffix trie. Laat zien dat het geheugenbeslag $\mathcal{O}(n)$ is, waar n de lengte van de gerepresenteerde string is. Illustreer een en ander aan de hand van de string 'abracadabras'.
5. Leg uit wat het beslissingsprobleem CIRCUIT-SAT inhoudt. Laat zien dat het in de complexiteitsklasse NP zit. Zit het ook in de klasse P?