Universiteit Twente
Afdeling Informatica, Faculteit EWI
11 april 2008, 13:30 – 17:00

Semester 2007-2008/1
Tentamen

Algoritmen, Datastructuren en Complexiteit

(214020 en 214025)

Bij dit tentamen mag het boek van Baase en Van Gelder worden gebruikt, evenals een uitdraai van de hoorcollegesheets (dit alles zonder eigen aantekeningen).

Bij de opgaven waar om een algoritme wordt gevraagd, geeft u de pseudocode van uw oplossing en een beknopte maar duidelijke uitleg van de werking. Algoritmes zonder duidelijke uitleg worden niet in beschouwing genomen.

Uitspraken die u doet in antwoord op gestelde vragen moeten nauwkeurig worden beargumenteerd.

Er zijn 5 opgaven, waarmee 90 punten behaald kunnen worden. Het tentamenresultaat is (het aantal behaalde punten gedeeld door 10) plus 1.

Vermeld uw naam en de afkorting ADC op ieder los blad. Vermeld ook de werkcollegeleider waar u dit jaar bij was ingedeeld en geef expliciet aan of u beide huiswerkopgaven gemaakt hebt.

Veel succes!

Opgave 1

`15 pt`

Beschouw het volgende algoritme (met `*` vermenigvuldigen, `div` integer division (bv. `7 div 2 = 3`), en `^2` kwadraat):

```
int func(int n)
{ if n == 0 return 1
    else if n < 8 return n
       else return 3*func(n div 8) + 8 + func(n div 8)^2
}
```

1. Geef een recursieve uitdrukking van de tijdscomplexiteit van dit algoritme, uitgedrukt in het aantal rekenkundige operaties.

2. Wat is de complexiteitsklasse van dit algoritme?

Opgave 2

`20 pt`

Stel we hebben een maxheap met n verschillende elementen (gegeven als een array) en $n > 2$. We weten toevallig dat het kleinste element van deze heap index k heeft.

1. Geef een zo efficient mogelijk algoritme (qua aantal vergelijkingen op elementen) dat het op een na kleinste element van de heap bepaalt.

2. Wat is de complexiteit van dit algoritme (dus niet de complexiteitsklasse), waarbij we het aantal gemaakte vergelijkingen tellen?

Opgave 3

20 pt

Zij $G = (V, E)$ een gerichte graaf.

1. Geef een DFS algoritme dat bepaalt of G een cykel heeft.
2. Geef de worst-case complexiteit van dit algoritme.

Opgave 4

25 pt

Beschouw het volgende spel. Het spel wordt gespeeld op een bord met n bij n vierkante vakjes. Je mag een damsteen op een willekeurig vakje op de onderste rij zetten. De damsteen mag je vervolgens steeds diagonaal linksomhoog of rechtsomhoog schuiven, mits je op het bord blijft. Voor elke zet kun je een bepaald positief aantal punten krijgen, die in een tabel gegeven zijn: vanuit vakje (i, j) rechtsomhoog levert $p(i, j, R)$ punten op, linksomhoog levert $p(i, j, L)$ punten op. Uiteindelijk mag je op een willekeurig vakje op de bovenste rij eindigen. Neem aan dat het vakje linksonder de coordinaten $(1, 1)$ heeft.

1. Geef een recurrente betrekking voor het maximaal aantal punten bij aankomst in vakje met coordinaten (i, j).
2. Geef een algoritme om te bepalen hoeveel punten je maximaal kunt winnen. De complexiteit mag niet slechter zijn dan kwadratisch in n.
3. Geef aan hoe het algoritme zo aangepast kan worden dat ook het beginvak en de winnende zettenreeks geproduceerd kunnen worden.

Opgave 5

10 pt

Geef van de volgende beweringen aan of ze waar of onwaar zijn, en motiveer je antwoord.

1. Beschouw de recurrente betrekking $T(n) = 2 \cdot T(\frac{n}{2}) + 2 \cdot log\ n$, $T(1) = 1$. Volgens het Masters theorema geldt $T(n) \in \Theta(log\ n)$.
2. Stel je gebruikt gesloten hashing. Als de gemiddelde lijstlengte 5 is, en er 5000 elementen zijn, is de loadfactor kleiner dan 1.
3. Als je het travelling salesman probleem polynomiaal zou kunnen reduceren tot een ander probleem, heb je bewezen dat P = NP.
4. Als we zouden kunnen bewijzen dat P ongelijk is aan NP, dan hadden we tevens bewezen dat problemen in NP niet efficient opgelost kunnen worden.