

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

Beschouw het volgende algoritme (met * vermenigvuldigen, div integer division (bv. $7 \text{ div } 2 = 3$), en $\wedge 2$ kwadraat):

15 pt

```
int func(int n)
{ if n == 0 return 1
  else if n < 4 return n
    else return 2*func(n div 4) + 6 + func(n div 4)^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

1. Geef een zo efficiënt mogelijk algoritme dat het verschil bepaalt tussen het grootste en het kleinste element van een heap (gegeven als een array).
2. Wat is de complexiteit van dit algoritme (dus niet de complexiteitsklasse), waarbij we tellen het aantal gemaakte vergelijkingen?

20 pt

Opgave 3

20 pt

Zij $G = (V, E)$ een gerichte graaf met een centrale knoop c van waaruit alle andere knopen bereikbaar zijn. G heet *equidistant* als voor elke knoop v geldt dat alle paden van c naar v dezelfde lengte hebben. Geef een DFS algoritme dat bepaalt of G equidistant is. Wat is de worst-case complexiteit?

Opgave 4

25 pt

Beschouw het volgende spel. Het spel wordt gespeeld op een bord met n bij n vierkante vakjes. In elk vakje ligt een aantal rijstkorrels (minstens 1 per vakje). Je begint in het vakje linksonder, en mag steeds of een vakje naar boven of een vakje naar rechts. Uiteindelijk moet je in het vakje rechtsboven aankomen. In elk vakje dat je passeert (inclusief de vakjes van vertrek en aankomst) mag je de rijstkorrels in dat vakje pakken. Het doel van het spel is zoveel mogelijk rijstkorrels te verzamelen.

1. Geef een recurrente betrekking voor het aantal gewonnen rijstkorrels bij aankomst in vakje met coördinaten (i, j) .
2. Geef een algoritme om te bepalen hoeveel rijstkorrels je maximaal kunt winnen. De complexiteit mag niet slechter zijn dan kwadratisch in n .
3. Hoe kun je met dit algoritme ook de winnende zettenreeks produceren?

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 $T(n) = T(\frac{n}{2}) + 2 \cdot \log n$, $T(1) = 1$. Volgens Masters theorema geldt $T(n) \in \Theta(\log n)$.
2. Stel je gebruikt voor open adressering een hastabel met 400 locaties. Stel dat de kans, dat een item op een locatie wordt afgebeeld, voor alle items en alle locaties even groot is. De kans dat het vijfde toegevoegde item to een hashcollision leidt is 0,01.
3. Het Euler circuit probleem (heeft een graaf een cykel waarbij elke edge precies 1 keer bezocht wordt, terwijl knopen vaker mogen voorkomen) is polynomiaal reduceerbaar tot het Hamilton cykel probleem.
4. Een NP-volledig probleem is alleen efficiënt oplosbaar als het ook in P zit.