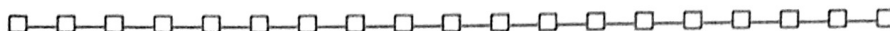


TENTAMEN ORIENTATIE INFORMATICA

10 februari 2011, 9:00-12:00 uur



*Dit tentamen is GEEN open boek tentamen. Je mag WEL één van de artikelen meenemen uit IEEE Computer Magazine danwel IEEE Computational Intelligence Magazine.*

*Voorzie de in te leveren bladen van je naam, en nummer ze. Schrijf op het eerste blad het aantal ingeleverde bladen. Met elk van de 11 vragen is 3 punten te verdienen. Je hoeft dus maar drie opgaven te maken. Als je meer maakt kijk ik slechts de eerste drie na, tenzij je een antwoord duidelijk doorstreept! Je krijgt 1 punt gratis. Succes!*

**Opgave 1. NOAG-ict**

De NOAG-ict bevat veel dwarsverbanden tussen de thema's, deels omdat verschillende vakgebieden in meerdere thema's aan de orde komen, deels omdat ze bij dezelfde maatschappelijke ontwikkelingen van belang zijn. Kies twee van de NOAG-ict thema's, en schrijf een essay van 350-400 woorden, waarin je in je eigen woorden uitlegt wat de dwarsverbanden

**Opgave 2. Eén van de artikelen waar je geen essay over hebt geschreven!!**

Neem het artikel dat je bij je hebt *maar waar je geen essay over hebt ingeleverd!!* Schrijf een samenvatting van 350-400 woorden, waarin je de belangrijkste stellingnames verwoord, en (met name als er veel voorbeelden zijn) de meest aansprekende voorbeelden kort beschrijft.

**Opgave 3. Geschiedenis**

Alan Turing bedacht in de jaren 30 van de vorige eeuw het concept van de Turingmachine.

- a. Geef een omschrijving van dit concept, i.h.b. de structuur en werking van de machine.
- b. Geef aan waarin dit concept afwijkt van de mechanische rekenmachines die in die tijd al bestonden.
- c. Geef de relatie aan tussen de Turingmachine en de huidige computer. Ga hierbij in op aspecten van de huidige computer waarvan de Turingmachine abstraheert.

**Opgave 4. Software Engineering**

Schrijf in een kort opstel van maximaal 350-400 woorden op wat voor Software Engineering gerelateerde baan je na je studie zou willen hebben. Beschrijf kort in welk domein (b.v. computer games, telecom, financiële, industrie, etc.) je deze baan zou willen uitvoeren. en in wat voor soort functie (b.v. consultant, programmeur, architect, tester, onderzoeker, eigen bedrijf etc.). Probeer een duidelijke relatie tussen deze baan en Software Engineering (SE). Geef hierbij antwoord op de volgende twee vragen: Wat zullen de (SE) problemen zijn waar je mee te maken krijgt en wat zijn de uitdagingen in dit domein m.b.t. SE?

**Opgave 5. Gedistribueerde systemen**

Beschouw het systeem van geldautomaten van de Europese banken als een gedistribueerd systeem. Schrijf een essay van 350-400 woorden over de uitdagingen in het ontwerpen en onderhouden van een dergelijk gedistribueerd informatie systeem (b.v., heterogeneity, scalability, openness, etc.). Geef van ieder term die je gebruikt ook aan wat het betekend en maak een afweging van het belang in deze context (b.v. door ze te prioriteren).

**Opgave 6. Parallel rekenen**

Cruciaal bij het behalen van hoge snelheden bij parallel rekenen is de architectuur van de interconnecties tussen de processing nodes. Geef een beschrijving van 4 architecturen (b.v. bus, crossbar switch, complete communication graph, en hypercube(beperk je bij de laatste tot b.v. 8 nodes)) met de voor- en nadelen.

**Opgave 7. Non-photorealistic rendering**

In computer graphics was in het algemeen het doel om beelden te maken die zo goed mogelijk op echte

foto's lijken. Pas recent zijn onderzoekers technieken gaan ontwikkelen die zich van dit ene doel "losweken" en hebben het vakgebied "non-photorealistic rendering" ontwikkeld.

Bespreek in 350-400 woorden wat deze nieuwe technieken te bieden hebben, wat toepassingsgebieden zijn, en welke uitdagingen er zijn die niet in de traditionele computer graphics te vinden zijn. Bespreek ook hoe interfaces een rol spelen binnen de computer graphics en in het creëren van non-fotorealistische beelden.

### Opgave 8. Visualisatie

Visualisatie van grote data sets is een ingewikkeld probleem, en er zijn vele oplossingen voor bedacht. Kies een aantal voorbeelden gegeven op het college, en geef in een essay van 350-400 woorden in grote lijnen aan wat ze doen, en waarom ze voor bepaalde toepassingen geschikt zijn. Neem hierbij tenminste één voorbeeld uit de "scientific visualisation" en één uit de "information visualisation". Geef aan wat deze termen betekenen en hoe de visualisatie technieken verschillen.

### Opgave 9. Computer Vision

In het volgende mag je uitgaan van binaire beelden. Ieder onderdeel 1 punt

- In connected filtering spelen z.g. "connected components" (samenhangende componenten) een belangrijke rol. Leg in je eigen woorden uit wat een connected component is. Teken een simpel voorbeeld waarin je aangeeft wat je bedoelt.
- Attribuut-filters behouden of verwijderen connected components van een invoerbeeld op basis van de eigenschappen (attributen) van iedere component. Als ik van te voren niet weet hoe ver objecten van een camera af staan, of wat voor een lens ik voor de camera heb, hoe kan ik dan toch een attribuut filter gebruiken (hint: wat voor een soort eigenschappen moet ik dan gebruiken)? Geef een mogelijke toepassing.
- Het is mogelijk om z.g. contracterende connectiviteit (soms ook partitionerende connectiviteit genoemd) te gebruiken in attribuut filters. Leg uit wat dit doet, en wanneer dit nuttig is. Geef ook een toepassing als voorbeeld.

### Opgave 10. Wederzijdse uitsluiting

Er zijn een aantal processen die tegelijk een niet-eindigende lus uitvoeren:

```
while (true) {
    NCS // andere activiteit
    intro
    CS // critische sectie
    exit
}
```

Het probleem van wederzijdse uitsluiting is de commando's intro en exit zo te programmeren dat er nooit meer dan één proces tegelijk in CS is. We mogen hiertoe variabelen gebruiken die door alle processen gelezen en geschreven worden.

a) Beschrijf één of enkele alinea's het belang van wederzijdse uitsluiting voor computerprogramma's en computersystemen.

b) Over implementaties.

Bij het toilet gebruikt men gewoonlijk een deur die van binnen op slot gedaan kan worden. We kunnen dit modelleren met een gedeelde variabele

```
boolean open = true ; // initieel

while (true) {
    NCS // andere activiteit
    while (! open) skip ; // wacht tot de deur open is
    open = false ; // doe de deur achter je op slot
```

```

    CS // critische sectie
    open = true ;           // doe de deur weer open
}

```

Dit werkt helaas niet voor processen, want twee of meer processen zouden net na elkaar `open == true` kunnen lezen en dan allebei na elkaar `open = false` zetten en CS in kunnen gaan. Wel correct is:

```

while (true) {
    NCS // onbelangrijk
wacht: < if (! open) goto wacht ; else open = false ; >
    CS // critische sectie
    open = true ;
}

```

De hoekjes `<S>` om een commando `S` drukken uit dat het commando `S` atomair uitgevoerd moet worden, dus zonder interferentie van andere processen. De opdracht "goto wacht" is een sprongopdracht, zoals we die in ons programmeeronderwijs jullie nooit leren (wacht is het label waarnaartoe gesprongen wordt).

**b1)** Leg uit waarom dit proces niet eerlijk is.

Om oneerlijkheid te voorkomen voert men (bv) bij de apotheek een tellertje in. Dit is een gedeelde variabele

```
int teller = 0 ; // initieel
```

die in één atomaire opdracht gelezen en opgehoogd kan worden volgens

```
< eigenTeller = teller ; teller = teller + 1 ; >
```

**b2)** Programmeer nu eerlijke wederzijdse uitsluiting met deze teller. Zorg dat elk proces keurig op zijn beurt wacht. Argumenteer dat aan wederzijdse uitsluiting voldaan wordt en dat elk wachtend proces ooit aan de beurt komt. Je mag zelf desgewenst andere gedeelde variabelen en privévariabelen invoeren (maar houd het eenvoudig). Zorg dat alle variabelen goed geïnitieerd zijn.