Tentamen IN3105
Complexiteitstheorie

25 juni 2012,

9.00 – 12.00 uur

- Dit tentamen bestaat uit 10 meerkeuzevragen, 5 korte (open) vragen en 2 open vragen.
- Per meerkeuzevraag kunnen 0 tot 4 alternatieven juist zijn.
- Voor de meerkeuzevragen kunt u maximaal *50* punten behalen, voor de korte open vragen maximaal *30* punten en voor de open vragen maximaal *20* punten.
- Het totaal aantal punten voor de meerkeuzevragen (*mp)* wordt als volgt bepaald:
  - per meerkeuzevraag *i* wordt het aantal foutief beantwoorde alternatieven f_i bepaald (*0-4*);
  - de som $f = \sum f_i$ voor alle 10 meerkeuzevragen wordt bepaald $(0 \leq f \leq 40)$;
  - het aantal punten *mp* is dan gelijk aan $mp = 50 \times max\{ 0, (40 - 1.5 \times f) / 40 \}$.
- Het eindcijfer *c* wordt bepaald door de som te nemen van *mp,* het totaal aantal punten behaald voor de 5 korte vragen *kv* en de som van het aantal punten behaald voor de twee open vragen *ov*, en vervolgens het resultaat te delen door 10: *cv = (mv+kv+ov)/10;* dit cijfer wordt vervolgens afgerond op het dichtstbijzijnde halve of gehele getal.

- Het gebruik van dictaat, aantekeningen, boek of andere bronnen is niet toegestaan.
- Het gebruik van grafische en niet-grafische rekenmachines is eveneens niet toegestaan.
- Beantwoord de meerkeuze vragen op het bijgeleverde antwoordformulier.
- Beantwoord de open vragen zo bondig mogelijk; *geef geen irrelevante informatie, dit kan leiden tot puntenaftrek!*
- Controleer of u op ieder blaadje uw naam en studienummer heeft vermeld en geef het totaal aantal ingeleverde bladen op (tenminste) de eerste pagina.

Notationele conventies:

- $A \leq B$ betekent dat er een polynomiale-tijd reductie van A naar B bestaat.
- A^c is het complement van het probleem A.
- Er wordt steeds vanuit gegaan, dat $\mathbf{P} \neq \mathbf{NP}$ en $\mathbf{NP} \neq \mathbf{coNP}$, tenzij uitdrukkelijk het tegendeel wordt vermeld.

MEERKEUZEVRAGEN (50 PUNTEN)

Vraag 1 Als zou blijken dat **P** ≠ **NP**, dan zou gelden:

a. niet voor ieder **NP**-probleem bestaat er een exponentieel algoritme.
b. sommige **NP**-problemen kunnen in polynomiale tijd worden opgelost.
c. **NPO** ≠ **PO**.
d. voor elk tweetal **NP**-volledige problemen A en B geldt A ≤ B en B ≤ A.

Antwoord: a. is onjuist: NP is een subset van EXP;
b. is juist, want NP omvat P en P is niet leeg.
c. juist.
d. juist, dit geldt altijd.

Vraag 2. A, B en C zijn drie beslissingsproblemen, waarbij gegeven is dat B een **NP** probleem is. Als A ≤ B en B ≤ C, dan geldt:

a. Als A een sterk **NP-volledig** probleem is, dan is C ook sterk **NP-volledig**.
b. Als C ∈ **NP**, dan A ∈ **P.**
c. Als A ∈ **coNPC**, dan C ∈ **coNP-hard**.
d. A ∈ **NP**.

Antwoord a. is onjuist, neem maar eens SAT ≤ VC ≤ SUBSETSUM
b. onjuist, want A, B en C kunnen NP-volledig zijn.
c. juist, want er is een polynomiale reductie van een coNPC-probleem naar C.
d. juist, want NP is naar beneden gesloten onder ≤.

Vraag 3. Voor de klasse $\mathbf{P}^{\mathbf{NP}}$ geldt:

a. Deze klasse is identiek aan de klasse $\mathbf{NP}^{\mathbf{P}}$.
b. Deze klasse is gelijk aan de klasse $\mathbf{P}^{\mathbf{coNP}}$.
c. Deze klasse omvat de klasse **coNP**.
d. Elk probleem in **NP** is polynomiale-tijd reduceerbaar tot een probleem in $\mathbf{P}^{\mathbf{NP}}$.

Antwoord a. is onjuist: $NP^P = NP$ is bevat in P^{NP}.
b. juist
c. juist
d. juist.

Vraag 4. Men wil nagaan of in een array A van *n* getallen er een getal is dat meer dan *n/2* maal voorkomt. Het volgende randomized algoritme wordt hiervoor gebruikt.

```
majority(A,n)                 % A is een array met n elementen
    begin
       index:= random(1,n); % kies een random getal uit 1..n
       x := A[index];
       count := 0;
       for k=1 to n do
            if A[k]=x then count:=count+1;
            k:=k+1;
       return (count>n/2);
     end
```

Er geldt nu dat:

a. Dit algoritme in *O(n)*-tijd altijd een correct antwoord geeft als A een waarde bevat die meer dan *n/2*-maal voorkomt.
b. voor elke positieve integer *k* er een randomized algoritme bestaat met polynomiale looptijd dat met kans $< 0.5^k$ een incorrect antwoord op het bovenstaande probleem geeft.
c. de kans dat dit algoritme ten onrechte concludeert dat A een waarde bevat die meer dan *n/2*-maal voorkomt, gelijk is aan 0.
d. dit probleem tot de klasse **BPP** behoort.

Antwoord a. is onjuist: met kans < 0.5 kan er een getal gekozen worden dat minder dan n/2 keer voorkomt
b. juist: herhaal bovenstaand algoritme *k*-maal en retourneer true als tenminste eenmaal true wordt opgeleverd.
c. juist
d. juist: er bestaat een algoritme (zie alternatief b en kies $k = 2$) zodat de kans op een false positive 0 is en de kans op een false negative kleiner dan 1/3.

Vraag 5. Het FALSESAT probleem is het probleem om, gegeven een verzameling *U* van propositiesymbolen en een verzameling *C* van clauses over *U,* te beslissen of er een waarheidstoekenning τ bestaat die *tenminste één* clause in C *onwaar* maakt.
Dit probleem is een

a. **NP**-probleem.
b. **NPC-**probleem.
c. **P-**probleem.
d. **coNPC**-probleem.

Antwoord: Zo'n waarheidstoekenning bestaat als er minstens één clause is waarin niet zowel een literal *x* als zijn complement $\neg x$ voorkomt. Zoiets is te checken in O(|C|) tijd. Het is dus een P-probleem
a. is juist: NP omvat P
b. is onjuist
c. is juist
d. is onjuist

Vraag 6. Het probleem om, gegeven een ongerichte graaf $G = (V, E)$, te bepalen of G minstens één, maar ten hoogste twee verschillende hamiltoonse paden heeft, is een probleem in

a. **NP**.
b. **co-NP**.
c. $\mathbf{P}^{\mathbf{NP}}$.
d. **NP ∩ co-NP**.

Antwoord: Dit probleem kun je oplossen door een NP-orakel twee vragen te stellen: heeft G een hamiltoons pad en heeft G niet meer dan 2 verschillende hamiltoonse paden?

a. is onjuist (tweede vraag behoort bij coNPC probleem)
b. is onjuist (eerste vraag hoort bij NP probleem)
c. is juist.
d. is onjuist. (dan zouden immers a. en b. ook juist zijn)

Vraag 7. Het probleem om, gegeven een graaf $G = (V, E)$ te bepalen of *G geen* vertex cover heeft die meer dan de *helft* van het aantal knopen in V bevat, is een

a. **NP**-volledig probleem.
b. **coNP**-volledig probleem.
c. **P**-probleem.
d. probleem in **NP ∩ co-NP**.

Antwoord Merk op dat V altijd een vertex cover is van G. Het probleem is derhalve in polynomiale tijd te beantwoorden met "nee".
a. is onjuist;
b. is onjuist;
c. is juist.
d. is juist, want P is een subset van **NP ∩ co-NP.**

Vraag 8. Voor een probleem P neemt men een approximatie algoritme A. Het is bekend, dat de performance ratio van A voor een zekere instantie x van het probleem gelijk is aan 1.5. Hieruit kan worden afgeleid dat:

a. A een 1.5-approximatie algoritme voor P is.
b. A geen c-approximatie algoritme voor P kan zijn voor een $c < 1.5$.
c. er geen 1.2-approximatie algoritme voor P kan bestaan.
d. P ∈ **APX.**

Antwoord a. is onjuist; uit instantie valt niet af te leiden wat geldt voor alle instanties
b. is juist;
c. is onjuist: uit dit ene algoritme kan dit niet afgeleid worden.
d. is onjuist (zie ook a.)

Vraag 9. Welke van de volgende uitspraken is waar :

a. SAT ≤ 3SAT.
b. VERTEX COVER ≤ PARTITION.
c. A ≤ SUBSETSUM voor een willekeurig **P**-probleem A.
d. A ≤ B en B ≤ A als A en B beide **NP**-harde problemen zijn.

Antwoord a. is juist; beide zijn NPC problemen
b. is juist; beide zijn NPC problemen
c. is juist: SUBSETSUM is een NPC probleem en P is bevat in NP.
d. is onjuist

Vraag 10. Welke van de volgende uitspraken is juist?

a. **PSPACE** $\subseteq$ **EXPTIME**.
b. **NPSPACE** $= \cup_{k \geq 0}$ **SPACE** (n^k).
c. **EXPTIME** $\subseteq$ **NPSPACE**.
d. **P** $\subsetneq$ **EXPTIME**.

Antwoord a. is juist;
b. is juist; er geldt PSPACE =NPSPACE
c. is onjuist.
d. is juist : volgt uit tijdhierarchiestelling

OPEN VRAGEN I (30 PUNTEN)

Onderstaande vragen dient u kort te beantwoorden.
Een antwoord dient in hoogstens 5-7 regels gegeven te worden.

Vraag 11. Waarom geldt **NPC** $\neq$ **NP**, ongeacht of **P=NP**, dan wel **P** $\neq$ **NP**?

Antwoord: $\varnothing$ en $\sum^*$ zijn P-problemen die niet volledig kunnen zijn.

Vraag 12. Waarom is het volgende probleem een **NP-hard** probleem:

MINIMUM 2-LEAF SPANNING TREE
Instantie: Een ongerichte graaf $G = (V, E)$
Vraag: Is er een opspannende boom T voor G met precies 2 bladeren?

Antwoord: Zo'n opspannende boom is een hamiltoons pad in G.

Vraag 13. Van **NL**, de klasse van problemen die in niet-deterministisch logaritmische ruimte zijn op te lossen, is bekend dat **L** $\subseteq$ **NL** $\subseteq$ $\mathbf{L}^2$.

Laat zien dat **NL** bevat is in **P**.

Antwoord: Omdat NL $\subseteq$ L^2 kan een Tm T die een taal in NL beslist, voor een invoer x in hoogstens $2^{\log p(|x|)^{\wedge}2} = q(|x|)$ configuraties verkeren voor een polynoom $q()$. Dat betekent dat de Tm in polynomiale tijd een beslissing moet nemen.

Vraag 14. Stel dat **P = NP**. Laat zien dat er dan een polynomiaal algoritme bestaat om een hamiltoons pad in een graaf te *vinden.*

Antwoord: HAMPAD is nu een P-probleem. Roep eerst HAMPAD aan voor G. Als "no", dan retourneer nil: er is geen hampad. Anders: neem een kant $\{x,y\}$ uit de graaf G . Maak een nieuwe graaf G' waarin x en y tot een knoop zijn versmolten en roep HAMPAD aan voor G'. Als "yes" dan behoort $\{x,y\}$ tot hamiltoons pad en herhaal procedure met G'. Anders: kies een andere kant $\{x', y'\}$ totdat er een kant op het hamiltoonspad wordt gevonden en herhaal procedure zoals beschreven. Deze procedure is polynomiaal en doet aanroepen naar een polynomiaal algoritme.

Vraag 15. Welk voorbeeld(en) kun je noemen om aan te geven dat met quantum computing geen exponentiële speed-up van algemene zoekmethoden voor NP-complete problemen te verwachten is?

Antwoord: Grovers's algoritme en de resultaten van Berstein-Vazirani laten zien dat algemene zoekmethoden met quantum computing hoogstens een kwadratische speed-up zullen geven.

OPEN VRAGEN II (20 PUNTEN)

Beantwoord onderstaande vragen zo kort mogelijk. Geef geen irrelevante informatie, dit kan tot puntenaftrek aanleiding geven.

Vraag 16. *(5 punten)*

Dr. ir. Ritant beweert dat LANGSTE PAD een **P**-probleem is. Zijn redenering is als volgt:

> Een instantie I van dit probleem bestaat uit een graaf $G = (V,E)$ en een integer $K \geq 1$, en om de vraag te beantwoorden of G een simpel pad ter lengte van K bevat, kunnen we het volgende algoritme toepassen:
>
> 1. Selecteer alle deelverzamelingen E' van E die precies K kanten bevatten;
> 2. Ga voor iedere geselecteerde deelverzameling E' na of uit de K kanten een simpel pad ter lengte van K te vormen is;
> 3. output "yes" als er minstens één E' te vinden is waaruit zo'n simpel pad ter lengte van K samen te stellen is; anders output "no".
>
> De looptijd van dit algoritme is polynomiaal: Er zijn hoogstens $O(|E|^K)$-deelverzamelingen E' van E ter grootte van K; per deelverzameling E' kost het niet meer dan $O(K^2)$-tijd om na te gaan of er een simpel pad ter lengte van K te vormen is uit de kanten van E'; het totale algoritme kost derhalve $O(|E|^K.K^2)$-tijd. Dit is, voor vaste K, een algoritme dat polynomiaal is in de grootte $|I| = |V| + |E| + \log K$ van de probleeminstantie I.
> Derhalve bestaat er een polynomiaal algoritme voor dit probleem.

Geef precies aan waarom deze redenering niet correct is.

Antwoord: Het algoritme kost $O(|E|^K.K^2)$-tijd en dat is *niet* polynomiaal in de grootte invoer $|I| = |V| + |E| + \log K$. Neem maar eens $K = O(|E|)$ en $|V| < |E|$. Dan geldt: $|I| = O(|E|)$ en de run-time van het algoritme is $O(|E||^{E|})$.

Vraag 17. *(15 punten)*

Neem het volgende probleem:

BUSROUTEPLANNING

Instantie: Een verzameling S van n steden, een matrix $D = [d_{ij}]_{nxn}$, $d_{ij} \in Z^+$, van intercity reistijden, een verzameling $H \subseteq S$ van haltes, een startplaats $s \in S$, een eindpunt $t \neq s$ in S, een verzameling van volgorde constraints $C \subseteq H \times H$ en een maximale tijdsduur T.

Vraag: Is er een route $R = (s_{i1}, s_{i2}, \ldots, s_{im})$ die alle haltes in H aandoet, startend vanuit $s = s_{i1}$ en aankomend in $t = s_{im}$, zodanig dat

- de totale reistijd van deze route niet meer dan T tijdseenheden bedraagt; dwz. $\Sigma_{k=1,\ldots m-1}\, d_{ik,i(k+1)} \leq T$;

- de volgorde constraints in C niet geschonden worden, dwz. als $(h_i, h_j) \in C$, dan mag h_j niet voorafgaan aan h_i in de route R van s naar t.

a. Laat zien dat dit probleem een **NP**-probleem is.

Antwoord: Gok een oplossing R en ga na
- of alle haltes in H worden aangedaan: kost $O(|H|) \leq O(|I|)$ –tijd.
- of de totale reistijd niet meer dan T bedraagt: kost $O(|H|.|D|+|H|.\ log\ max\{D\}) = O(|I|^2)$*-tijd*
- of aan alle constraints is voldaan: kost $O(|C|.|H|) \leq O(|I|^2)$*-tijd.*

b. Laat zien dat dit probleem een **NP-hard** probleem is.

Antwoord: We tonen aan dat TSP ≤ BUSROUTEPLANNING.

Neem een instantie *(S, D, B)* van TSP. Construeer de volgende instantie
$I = (S', s, t, D', C, H, T)$ van BUSROUTEPLANNING.
1. $S' = S \cup \{t\}$, t komt niet voor in S.
2. s is een willekeurige stad in S zeg s_1
3. $D' = [d_{ij}]$ is als volgt opgebouwd:
   - voor elke $i,j \in S$ geldt $d'_{ij} = d_{ij}$;
   - voor elke $i \neq s \in S$ geldt $d'_{it} = d_{is}$ *en* $d'_{ti} = d_{si}$;
   - voor $s \in S$ geldt $d'_{st} = d'_{ts} = B + 1$
4. C = ∅
5. $H = S$
6. $T = B$

Stel nu dat $P = (s, s_2 \ldots s_n, s)$ is een TSP-tour in *(S, D, B)* is met kosten ≤ B. Dan is $(s, s_2, \ldots s_n, t)$ een route van het geconstrueerde busrouteplanningsprobleem dat voldoet en ook kosten heeft ≤ $B = T$. Omgekeerd, stel, dat $(s, s_2, \ldots s_n, t)$ een oplossing is van BUSRTOUTEPLANNING met kosten ≤ T = B dan is $(s, s_2, \ldots s_n, s)$ een TSP-tour met kosten ≤ B. Immers $d'_{s\_n,t} = d_{s\_n,s}$.
De reductie kan in polynomiale tijd worden uitgevoerd.
Hiermee is aangetoond dat TSP ≤ BUSROUTEPLANNING en daarmee dat BUSROUTEPLANNING een NP-hard probleem is.

EINDE TENTAMEN