

Antwoorden opgaven werkcollege 10

Opgave 54

We vertalen het probleem naar het vijf-kleuren van een graaf, die we construeren als volgt.

- elke knoop stelt een commissie voor
- tussen een paar knopen loopt een tak dan en slechts dan als er een persoon is die vertegenwoordigd is in beide commissies, dus als de bijbehorende commissies overlappende leden hebben

Het probleem is dan om elke commissie (knoop) een middag (kleur) toe te wijzen op zo'n manier dat twee commissies met overlappende leden (naburige knopen) niet dezelfde middag (kleur) hebben.

Opgave 59

a. Zie uitwerking website

b. We geven een niet-deterministisch algoritme voor dit probleem in drie fasen. Globaal idee: gok een deelverzameling van S en controleer in Fase 2 of dit een gewenste deelverzameling is, dat wil zeggen of de som der elementen gelijk is aan t . Omdat we in Fase 1 een willekeurige string gokken, moet in Fase 2 ook gecontroleerd worden dat die string een deelverzameling van S voorstelt.

Het algoritme, met $x = \langle S, t \rangle$ de invoer van het probleem:

- Fase 1 (gokfase): Er wordt een string s gegenereerd, hierna te interpreteren als een rij gehele getallen.
- Fase 2 (verificatiefase): Er wordt gecontroleerd of s een deelverzameling van S is die tot t sommeert, door de volgende stappen:
 - controleer dat elk getal uit s in S voorkomt (elk getal s_i opzoeken in S : $O(|s||S|)$);
 - controleer dat elk getal in s precies één keer voorkomt ($O(|s|^2)$);
 - controleer dat de getallen uit s samen sommeren tot t (optellen en vergelijken met t : $O(|s|)^1$).

Als alle tests positief zijn, retourneer True; zodra aan een test niet is voldaan retourneer dan False (of ga in een oneindige loop, of het programma blijft hangen, ...).

- Fase 3 (uitvoerfase): Als Fase 2 True oplevert, voer dan **ja** uit; anders geen uitvoer ($O(1)$).

Nu geldt: het antwoord van het algoritme is **ja** $\Leftrightarrow$ er is een executie van het algoritme die **ja** oplevert (definitie antwoord niet-deterministisch algoritme)

¹Eigenlijk $O(|s| + |t|) \subseteq O(|s| + |x|)$, polynomiaal in de lengte van s en de lengte van de invoer x , en dus mogen we hier het vergelijken met t ook wel als 1 stap tellen.

$\Leftrightarrow$ er bestaat een string s waarvoor Fase 2 True geeft (structuur van het niet-deterministisch algoritme) $\Leftrightarrow$ er bestaat een string s die een deelverzameling van S voorstelt en die tot t sommeert (onze eigen invulling van Fase 2, waar we controleren of s een oplossing is) $\Leftrightarrow \langle S, t \rangle$ is een ja-instantie voor SubsetSum (definitie van het probleem).

Verder is het algoritme polynomiaal begrensd: als $\langle S, t \rangle$ een ja-instantie is, dan is er een executie die **ja** oplevert, en waarvoor de string s dus een goede deelverzameling van S voorstelt. Derhalve is dan $|s| \leq |S|$ en dus $|s| \in O(|x|)$. Fase 1 is dan $O(|x|)$, Fase 2 is $O(|x|^2)$ en in totaal is dat $|x|^2$, dus polynomiaal in de lengte/grootte van de invoer.

c. Het probleem Klik is als volgt: gegeven een graaf $\mathcal{G} = (V, E)$ en een geheel getal k ($0 \leq k \leq |V|$), bestaat er een deelverzameling V' van V met ten minste k knopen, waarbij elk tweetal knopen uit V' verbonden is in $\mathcal{G}$? De invoer van het probleem is dus $\langle \mathcal{G}, k \rangle$. We mogen wel aannemen dat de knopen worden aangegeven door gehele getallen, bijvoorbeeld dat de knoopverzameling $V = \{1, 2, \dots, |V| - 1, |V|\}$, met $|V|$ het aantal knopen van V . Verder nemen we aan dat E wordt weergegeven als een verzameling/rij ongeordende paren (u, v) (met $u, v \in V$).

We geven een niet-deterministisch algoritme:

- Fase 1 (gokfase): Er wordt een string s gegenereerd, te interpreteren als een rij gehele getallen die we beschouwen als knoopnummers.
- Fase 2 (verificatiefase): Er wordt gecontroleerd of s een klik met ten minste k knopen voorstelt. De eerste twee stappen controleren of s een deelverzameling van V voorstelt.
 - controleer dat elke knoop (getal) in s in V zit: $O(|s| \cdot |V|)$;
 - controleer dat s geen dubbele knopen bevat: $O(|s|^2)$;
 - controleer dat s ten minste k knopen bevat: tellen en vergelijken met k , dus $O(|s|)$, waarbij het vergelijken met k voor 1 stap is geteld;
 - controleer dat voor elke twee knopen u, v uit s met $u \neq v$, gezien als knopen in V , geldt dat er een tak is tussen u en v : alle tweetallen knopen uit s aflopen en in E kijken of er in $\mathcal{G}$ een tak tussen zit: $O(|s|^2 \cdot |E|)$.

Als alle tests positief zijn, retourneer True; zodra aan een test niet is voldaan retourneer dan False (of ga in een oneindige loop of blijf hangen of ...).

- Fase 3 (uitvoerfase): Als Fase 2 True oplevert, voer dan **ja** uit; anders geen uitvoer.

Nu geldt: het antwoord van het algoritme is **ja** $\Leftrightarrow$ er kan een string s gegokt worden waarop Fase 2 True geeft $\Leftrightarrow$ er bestaat een string s die een klik van $\mathcal{G}$ voorstelt met ten minste k knopen $\Leftrightarrow$ er bestaat in $\mathcal{G}$ een klik ter grootte ten minste k $\Leftrightarrow \langle \mathcal{G}, k \rangle$ is een ja-instantie is voor Klik.

Als s inderdaad een klik is, dan is het een deelverzameling van V , dus $|s| \leq |V|$ en derhalve $|s| \in O(|x|)$. In dat geval is de complexiteit van Fase 1 dus $O(|x|)$ en van Fase 2 $O(|V|^2 \cdot |E|) \subseteq O(|\mathcal{G}|^3) \subseteq O(|x|^3)$, en dat is polynomiaal. Er geldt dus dat als $\langle \mathcal{G}, k \rangle$ een ja-instantie is, dan bestaat er een executie die **ja** oplevert in $O(|x|^3)$ stappen; het niet-deterministische algoritme is dus polynomiaal begrensd.

Opgave 62

Zie uitwerking website

Opgave 63

a. Globaal idee: voeg een extra knoop toe die is verbonden met elke andere knoop. Deze moet voor een goede kleuring een andere kleur hebben dan alle andere knopen, dus deze graaf is 4-kleurbaar d.e.s.d.a. de rest van de graaf (dus de oorspronkelijke graaf) 3-kleurbaar is.

Iets preciezer: - de oorspronkelijke ongerichte graaf $\mathcal{G}$ wordt afgebeeld op een nieuwe graaf $\mathcal{G}'$ met één extra knoop en takken van die knoop naar alle andere knopen. Deze constructie is ten duidelijkste polynomiaal in $|\mathcal{G}|$. Geef zelf kort een algoritme voor deze omzetting, waarbij je bijvoorbeeld voor $\mathcal{G}$ en $\mathcal{G}'$ de adjacency list gebruikt. Je kunt de takkenverzameling E natuurlijk ook eenvoudig representeren als een rij van (ongeordende) paren knopen.

- Nu de reductie-eigenschap bewijzen, dat wil zeggen: $\mathcal{G}$ is een ja-instantie voor 3Kleur $\iff \mathcal{G}'$ is een ja-instantie voor 4Kleur.

" $\implies$ ": Stel dat $\mathcal{G}$ een ja-instantie is voor 3Kleur, d.w.z. $\mathcal{G}$ is 3-kleurbaar. Dan is $\mathcal{G}'$ 4-kleurbaar: we gaan uit van een correcte 3-kleuring van de knopen van $\mathcal{G}$ en kleuren de overeenkomstige knopen van $\mathcal{G}'$ zoals in die kleuring aangegeven. De onderlinge kleuring van die knopen is al goed. We kleuren de nieuwe knoop met de extra kleur. Deze is verbonden met alle andere knopen in $\mathcal{G}'$, maar heeft een andere kleur, dus dat is ook in orde.

" $\impliedby$ ": Andersom, stel dat $\mathcal{G}'$ 4-kleurbaar is, en neem zo'n 4-kleuring. De extra knoop is met alle andere knopen verbonden, en heeft *dus* een kleur die anders is dan die van de overige knopen; die overige knopen zijn correct gekleurd (want de 4-kleuring was een goede kleuring) met de drie overige kleuren. Dat geeft dus een 3-kleuring van de oorspronkelijke graaf $\mathcal{G}$.

b. We geven een niet-deterministisch algoritme. De invoer is een graaf $\mathcal{G} = (V, E)$. We mogen wel aannemen dat het aantal knopen in V bekend is (polynomiaal veel werk om dat te tellen, dus 'gratis' in de context van $\mathcal{NP}$) en dat $\mathcal{G}$ gerepresenteerd wordt met een adjacencylist. Het aantal knopen, dus de grootte van V , geven we even aan met $|V|$.

- (Fase 1: gokfase) Er wordt een string s gegenereerd, te interpreteren als een rij gehele getallen. Dit is geeft hopelijk een goede kleuring van de knopen aan.
- (Fase 2: verificatiefase) Er wordt gecontroleerd of s een geldige 3-kleuring van $\mathcal{G}$ voorstelt:

- controleer dat elk getal in s tussen 1 en 3 zit: $O(|s|)$;
- controleer dat het aantal getallen in s precies $|V|$ is: $O(|s|)$;

Als aan de vorige twee tests voldaan is stelt s een kleuring van de knopen voor met drie kleuren. We interpreteren s_i , het i -de getal uit s , als de kleur van de i -de knoop van V .

- controleer dat voor elke tak (v_i, v_j) in E geldt dat het i -de getal uit s ongelijk is aan het j -de getal in s (ergo $s_i \neq s_j$), en dus dat knopen die door een tak verbonden zijn een verschillende kleur hebben: $O(|s| \cdot |E|)$. Hier is het opzoeken van de kleur van knoop i in s als $O(|s|)$ geteld.

Als alle tests positief zijn, retourneer True, anders False of er wordt in een oneindige loop gegaan.

- (Fase 3: uitvoerfase) Als Fase 2 True oplevert, voer dan **ja** uit; anders geen uitvoer.

Nu geldt: het antwoord van het algoritme is **ja** d.e.s.d.a. er een string s bestaat/gegokt kan worden waarvoor Fase 2 True oplevert d.e.s.d.a. er een string bestaat die een 3-kleuring van $\mathcal{G}$ voorstelt d.e.s.d.a. $\mathcal{G}$ een ja-instantie is voor 3Kleur.

Als s inderdaad een 3-kleuring is, dan geldt $|s| = |V| \leq |\mathcal{G}|$. In dat geval is de complexiteit van Fase 2 dus $O(|V| \cdot |E|) \in O(|\mathcal{G}|^2)$, en dat is polynomiaal. Er geldt dus dat als $\mathcal{G}$ een ja-instantie is, dan bestaat er een executie die **ja** oplevert in $O(|\mathcal{G}|^2)$ stappen; het niet-deterministische algoritme is derhalve polynomiaal begrensd.

c. Uit de aanname ($3\text{Kleur} \in \mathcal{NP}$) volgt dat 3Kleur NP-hard is. Hieruit volgt samen met **a.** dat 4Kleur ook NP-hard is (transitiviteit van $\leq_P$). Om te concluderen dat 4Kleur NP-volledig is, moeten we echter ook nog weten dat $4\text{Kleur} \in \mathcal{NP}$, en dat is nog niet aangetoond. Het is overigens wel waar, want we kunnen voor 4Kleur een polynomiaal niet-deterministisch algoritme geven, geheel analoog aan dat voor 3Kleur.

Opgave 64

(1) Een ongerichte graaf is 1-kleurbaar d.e.s.d.a. hij geen takken heeft. Immers, als er een tak zou zijn, moeten de uiteinden van deze tak verschillende kleuren hebben en zijn er dus minstens 2 kleuren nodig. Als er geen takken zijn kunnen alle knopen dezelfde kleur hebben.

Een ongerichte graaf is 2-kleurbaar d.e.s.d.a. hij bipartiet² is, d.w.z. als de knopen zijn in te delen in twee deelverzamelingen zodat alle takken alleen tussen de twee deelverzamelingen lopen (en niet tussen paren knopen die in dezelfde deelverzameling zitten). Immers, als de knopen van de graaf met twee kleuren te kleuren zijn, zijn er twee deelverzamelingen van de knopen te onderscheiden, corresponderend met de twee kleuren. Elke tak moet dan lopen tussen een knoop uit de ene en een knoop uit de andere verzameling. En omgekeerd, als een graaf

²Zie ook het college Foundations of Computer Science

bipartiet is, kleuren we de knopen uit de ene deelverzameling blauw en die uit de andere rood. Dit is dan een goede kleuring.

(2) 1-kleurbaar: controleer of de graaf takken heeft. Zo ja, dan kan het niet. Zo nee, geef alle knopen dezelfde kleur. Deze controle kan in $O(|V|)$ stappen als $\mathcal{G} = (V, E)$ bij gebruik van de adjacencylist; in $O(|V|^2)$ als de adjacencymatrix wordt gebruikt. Dat is polynomiaal.

2-kleurbaar: neem als kleuren blauw en rood. Kleur een arbitraire knoop blauw. Kleur vervolgens alle burens rood. Bekijk vervolgens alle burens van de zojuist gekleurde knopen: als die blauw is dan is het tot nu toe goed; als die rood is, dan is het fout en dan kan het niet; als de buur nog ongekleurd is, dan kleur je deze blauw. Herhaal dit voor alle nieuw toegevoegde burens en met de juiste kleuren. Als je geen burens meer kan vinden maar nog wel knopen over hebt (de graaf is dan kennelijk niet samenhangend): kleur weer een arbitraire knoop blauw. Als dit proces termineert zonder fouten, heb je een juiste 2-kleuring gevonden. Zo niet, dan kan het niet. Merk op dat we hier eigenlijk een Breadth-First-Search hebben gedaan, en dit is ook polynomiaal ($\text{BFS kan in } O(|V|)+O(|E|) \subseteq O(|\mathcal{G}|)$).