

Alleen pen en papier toegestaan.

(Dus geen boeken, aantekeningen, rekenmachines en telefoons)

Voor vraag 1, 2 en 3 kunt u gebruik maken van het Antwoordblad.

Tentamen telt voor 75 % van eindcijfer. Huiswerk opdracht voor 25 %. Score per vraag:

1	2	3	4a	4b	4c	4d	4e	4totaal	5	6a	6b	6c	6totaal		Totaal
10	10	18	4	4	4	4	4	20	5	2	5	5	12		75

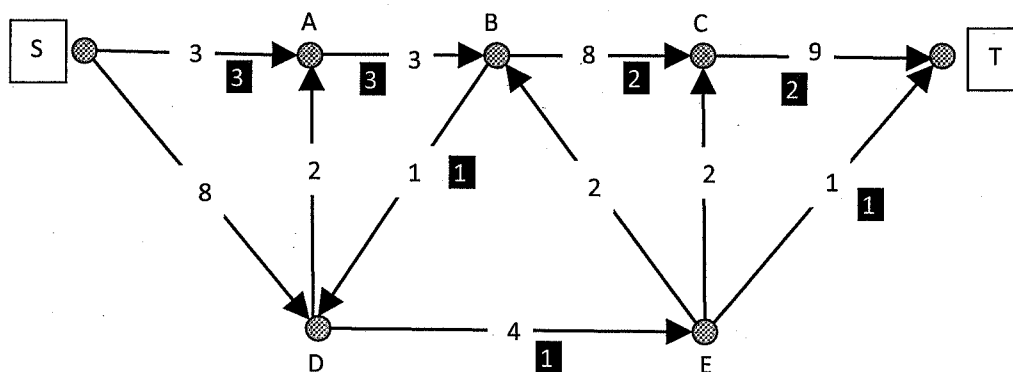
Vraag 1 Gegeven is de volgende matrix van afstanden tussen 4 punten. Vind de de lengtes van de kortste paden tussen elk tweetal punten door het Floyd-Warshall algoritme toe te passen. Neem als volgorde a,b,c,d.

C_{ij}	a	b	c	d
a	∞	3	4	1
b	6	∞	1	9
c	5	-1	∞	2
d	3	5	10	∞

VUL IN OP ANTWOODRBLAD

Vraag 2 Gegeven is het volgende netwerk met bron (source) S en put (sink) T. Op de pijlen staan de capaciteiten. Een beginstroom met waarde 3 is al aangegeven (zwarte vierkantjes). Vind de maximale S-T stroom uitgaande van de gegeven beginstroom door het Ford-Fulkerson algoritme toe te passen. Geef steeds de stroom in de figuur (stroom met waarde 0 hoef je niet aan te geven) en een stroomvermeerderend pad naast de figuur. Wat is de waarde van de maximale stroom die je vindt? Wat is de bijbehorende minimale snede?

VUL IN OP ANTWOODRBLAD.



Vraag 3. Geef van elke bewering of het goed of fout is. Toelichting is niet nodig. Een goed antwoord geeft +1. Een fout antwoord -1. Geen antwoord 0. ($0 \leq \text{score} \leq 18$.)

1. In een lineair programmerings probleem (LP-probleem) is elk lokaal optimum ook een globaal optimum.
2. Dijkstra's kortste pad algoritme loopt in $O(\log n)$ tijd.
3. Als in een gegeven netwerk alle capaciteiten geheel-tallig zijn dan zal de minimale s-t snede ook geheel-tallig zijn.
4. De waarde van de maximale s-t stroom is altijd gelijk aan de waarde van de minimale s-t snede.
5. Een lokaal zoek algoritme waarbij het doorzoeken van de neighborhood in polynomiale tijd gebeurt heeft een polynomiale looptijd.
6. Het vinden van een maximale stroom in een netwerk met onder- en bovengrenzen op de capaciteiten kan in polynomiale tijd.
7. Elke gewogen graaf heeft een unieke (dwz precies 1) opspannende boom van minimum gewicht.
8. List-Scheduling is een $6/5$ approximatie algoritme voor Load Balancing.
9. Het weighted vertex cover probleem is NP-moeilijk.
10. Voor elk optimaliserings probleem geldt dat geen enkel lokaal zoek algoritme gegarandeerd de optimale oplossing vindt. (Met andere woorden: elk lokaal zoek algoritme kan in een niet-optimaal lokaal optimum blijven steken.)
11. Een beslissingsprobleem dat NP-compleet is, is ook NP-moeilijk.
12. Het kortste pad probleem is NP-moeilijk.
13. $P \subseteq \text{co-NP}$
14. Laat Π_1, Π_2, Π_3 , beslissingsproblemen zijn. Als $\Pi_1 \leq \Pi_2$ en $\Pi_1 \leq \Pi_3$ dan ook $\Pi_2 \leq \Pi_3$.
15. Als G een ongerichte 3-reguliere graaf op 7 punten is dan is het complement van G ook 3-regulier.
16. Een ongerichte graaf waarin elk punt minstens graad k heeft, heeft een pad van lengte k .
17. Een ongerichte graaf met n punten en m lijnen ($n > m$) heeft maximaal $n-m$ componenten.
18. De complete bipartite graaf op 2+3 punten ($K_{2,3}$) is planair.

VUL IN OP ANTWOODRBLAD

Vraag 4. Laat $E = \{e_1, e_2, e_3, \dots, e_m\}$ een verzameling van m elementen zijn en laat $W = \{W_1, W_2, \dots, W_n\}$ een collectie van deelverzamelingen van E zijn. Een *set cover* S is een deelverzameling van W ($S \subseteq W$) zodat elk element van E in minimaal 1 verzameling van S voorkomt. De waarde van S is het aantal deelverzamelingen in S . Het *set cover probleem* is om gegeven E en W een set cover met kleinste waarde te vinden.

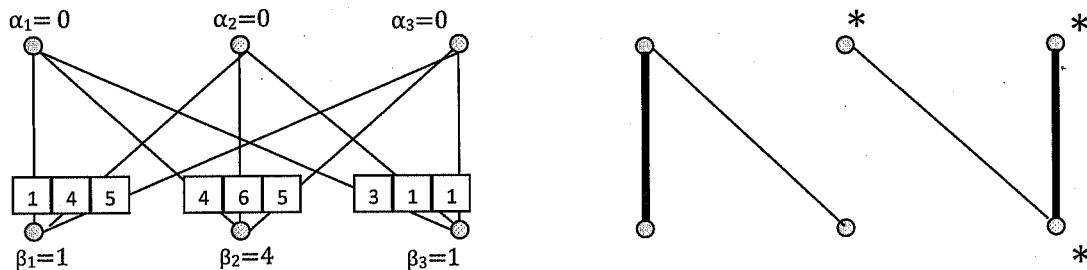
Voorbeeld:

Gegeven is $E = \{e_1, e_2, e_3, e_4\}$ en $W = \{W_1, W_2, W_3\}$, waarbij $W_1 = \{e_1, e_3\}$, $W_2 = \{e_2\}$ en $W_3 = \{e_1, e_2, e_4\}$. Een oplossing is $S = \{W_1, W_3\}$. De waarde van S is 2.

Het *vertex cover probleem* is het probleem om gegeven een (ongewogen en ongerichte) graaf $G = (V, E)$ een kleinste deelverzameling $S \subseteq V$ te vinden zodat elke lijn raakt aan minimaal 1 punt van S .

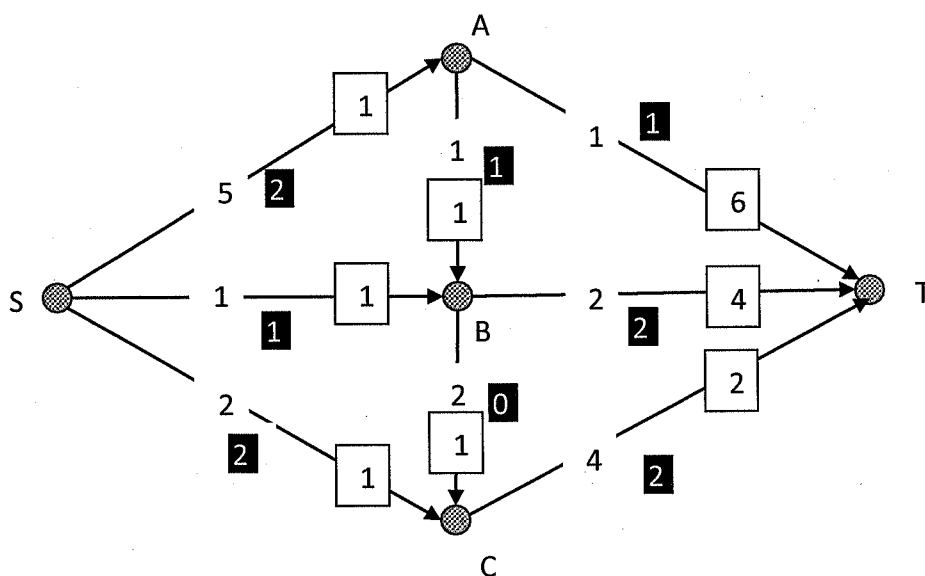
- Formuleer het vertex cover probleem als een set cover probleem.
- Geef een ILP formulering van het set cover probleem. Neem als variabele x_i , welke aangeeft of set W_i wel of niet in de cover zit.
- Hoe ziet de LP-relaxatie eruit?
- Beschouw een instantie van set cover waarbij elk element $e \in E$ in maximaal f verzamelingen van W voorkomt. (Waarbij $f \geq 2$ een constante is.) Beschouw het volgende algoritme:
 Los de LP-relaxatie op. Laat $(x_1^*, x_2^*, \dots, x_n^*)$ de gevonden oplossing zijn. Als $x_i^* \geq 1/f$ dan ronden we x_i^* af naar 1 en nemen we W_i in de set cover. Als $x_i^* < 1/f$ dan ronden we x_i^* af naar 0 en nemen we W_i niet in de set cover.
 Laat zien dat dit algoritme altijd een (toegelaten) set cover geeft.
- Bewijs dat de waarde van de bij e) gevonden oplossing altijd hooguit f maal de waarde van de optimale set cover is.

Vraag 5 Hieronder zie je een tussenoplossing van de Hongaarse methode. In de volgende stap zullen de duale variabelen weer aangepast worden. Wat worden de nieuwe waarden van de duale variabelen α_1 , α_2 , α_3 , en β_1 , β_2 , en β_3 ? (Je hoeft het niet verder af te maken.)



Vraag 6 In onderstaand netwerk N is een S, T - stroom f met waarde 5 aangegeven. Op elke pijl staat vermeld: de stroom (zwart vierkantje), de kosten per eenheid stroom (wit vierkantje) en de capaciteit (geen vierkantje).

- Wat zijn de totale kosten van deze beginstroom f ?
- Teken het *incremental flow network* $N'(f)$. (Ter herinnering: Hierin komt een pijl van punt i naar punt j als we de stroom van i naar j kunnen laten toenemen of als we de stroom van j naar i kunnen verminderen. Geef ook de kosten en resterende capaciteit bij elke pijl in $N'(f)$.)
- Heeft $N'(f)$ een negatieve kosten cycle? Zo ja, hoeveel stroom kun je toevoegen over deze cycle? Wat zijn de totale kosten van de nieuwe stroom als je deze toevoeging maakt?



Antwoordblad voor vraag 1,2,3. (Niet vergeten in te leveren)

Naam :

Studentnummer:

Vraag 1

C_{ij}	a	b	c	d
a	∞	3	4	1
b	6	∞	1	9
c	5	-1	∞	2
d	3	5	10	∞

d_{ij}	a	b	c	d
a				
b				
c				
d				

d_{ij}	a	b	c	d
a				
b				
c				
d				

d_{ij}	a	b	c	d
a				
b				
c				
d				

d_{ij}	a	b	c	d
a				
b				
c				
d				

Vraag 3

	goed	fout
1		
2		
3		
4		
5		

	goed	fout
6		
7		
8		
9		
10		

	goed	fout
11		
12		
13		
14		
15		

	goed	fout
16		
17		
18		

Vraag 2

