

Tentamen TI2306 Algoritmiek

1 februari 2017, 13.30-16.30

- Het gebruik van boek, aantekeningen of rekenmachine tijdens dit tentamen is **niet** toegestaan.
- Dit tentamen heeft 7 open vragen (en daarover verdeeld 65 punten) in totaal goed voor 6.5/10 van je cijfer en 15 meerkeuzevragen goed voor 3.5/10 van je cijfer. Merk op dat je eindcijfer voor dit vak ook voor $\frac{1}{3}$ uit het onafgeronde gemiddelde van het practicum bestaat (mits beide ≥ 5.0).
- Als je voor de huiswerkopgaven van dit collegejaar gemiddeld minstens een 5.8 hebt, dan wordt je cijfer voor dit tentamen met 0.5 verhoogd (maar nooit hoger dan 10.0).
- Voor de open vragen heb je ongeveer twee uur nodig:
 - Geef antwoord in correct Nederlands of Engels en schrijf leesbaar (gebruik eerst potlood of kladpapier).
 - Geef geen irrelevante informatie. Dit kan leiden tot puntenaftrek. Het aangegeven maximaal aantal regels is van toepassing op (getypte) regels en wordt dus aangepast aan de grootte van je handschrift en het gebruik van witruimte.
 - Als er wordt gevraagd om een efficiënt algoritme, geef dan het meest efficiënte algoritme dat je kunt bedenken. Een langzamer algoritme kan minder punten opleveren.
 - Als er wordt gevraagd om pseudocode, dan mag je daarin algoritmen uit het boek aanroepen, tenzij anders vermeld.
 - Als er wordt gevraagd om een bewijs, dan mag je daarin een stelling uit het boek gebruiken, tenzij het gevraagde juist nodig is voor het bewijs van die stelling.
 - Voordat je je antwoorden inlevert, controleer of op ieder blaadje je naam en studienummer staat en geef de vakcode en het aantal ingeleverde bladen voor de open vragen aan op (tenminste) de eerste pagina.
- Wat betreft de meerkeuzevragen:
 - Er is voor iedere vraag telkens maar één goed antwoord mogelijk.
 - Alle vragen tellen even zwaar, maar bij de puntentoekenning wordt wel gokkanscorrectie toegepast. Als je geen enkel antwoord geeft, wordt dit altijd fout gerekend.
 - Schrijf je antwoorden eerst op kladpapier en neem ze later over op het antwoordformulier.
 - Vul je studienummer zowel met cijfers in als met streepjes/blokjes.
 - Vul het antwoordformulier bij voorkeur in met pen (geen rode pen) of potlood (goed zwart maken); gebruik geen doorhalingen.
 - Probeer in totaal niet meer dan een uur aan de meerkeuzevragen te besteden.
- De tentamenstof bestaat uit Chapters 4–7 uit Algorithm Design van Kleinberg en Tardos (behalve secties 4.9*, 5.6, 6.10*, 7.4*, en 7.13*), en bovendien de notities over de Master Method en over bewijstechnieken (Proving guide), ook toegepast op eenvoudige graafalgoritmen (Chapter 3).
- Totaal aantal pagina's (zonder dit voorblad): 11.

Open vragen

1. (10 punten) Gegeven is een gerichte graaf $G = (V, E)$. Bewijs dat als er geen cycles in G zitten, dat er dan altijd een knoop is zonder inkomende kanten (in 10 regels).

Denk hierbij aan de volgende *tips voor een goed bewijs*: 1. Geef aan welke bewijstechniek(en) je gebruikt. 2. Leg zonodig notatie uit. 3. Maak aannames expliciet. 4. Verwijs als je een stelling gebruikt.

Antwoord: Dit is het bewijs voor Eq.3.19 uit het boek (p.102). Indien niets vermeld, ken 1 punt toe aan ieder van de volgende onderdelen.

- We geven een bewijs uit het ongerijmde.
- Laat een willekeurige gerichte graaf $G = (V, E)$ zonder cycles gegeven zijn.
- Stel dat iedere knoop een inkomende kant heeft.
- Kies een willekeurige knoop $v \in V$. Deze knoop heeft tenminste één inkomende kant (u, v) . Volg deze inkomende kant terug naar de knoop u .
- Herhaal dit voor knoop u ipv v . We kunnen dit onbeperkt volhouden. Maar na $|V| + 1$ stappen hebben we tenminste één knoop tweemaal bezocht. (3 punten)
- Laat C de serie knopen zijn tussen deze twee bezoeken. C is dan een cycle.
- Dit is in tegenspraak met het feit dat G geen cycles heeft.
- Dus niet iedere knoop heeft een inkomende kant.

Kan ook met contrapositie, dan te bewijzen: als alle knopen inkomende kant hebben, dan is er een cycle.

- Inductie: 2 of 3 punten.
- DAG: topologisch: max 3 punten.

2. Stel dat je n taken hebt en één machine. Ieder van de taken i heeft t_i tijd nodig op de machine. De machine kan slechts één taak tegelijk afhandelen. Het doel is om voor iedere taak i de eindtijd (completion time) c_i te bepalen zodat de som van de eindtijden ($\sum_i c_i$) zo klein mogelijk is. Bijvoorbeeld, eerst taak $t_1 = 4$ en dan $t_2 = 6$, geeft eindtijden $c_1 = 4$ en $c_2 = 10$ en dus een som van $4 + 10 = 14$.

- (a) (6 punten) Geef een greedy algoritme dat voor een serie van n taken het rooster voor de machine bepaalt dat de som van de eindtijden minimaliseert. Bepaal in het algoritme ook de eindtijden $c_1, c_2, \dots, c_n$. Licht je algoritme kort toe.

Antwoord: Beschouw het volgende algoritme.

Sorteer de taken oplopend op benodigde tijd t_i (hernummer de indices)

$c_0 := 0$

for $j := 1$ **to** n **do**

$c_j := c_{j-1} + t_j$

end

Ken (maximaal) 3 punten toe aan een correcte volgorde en 3 aan het correct berekenen van de eindtijden.

- max. 4 punten voor kwadratisch algoritme

- (b) (2 punten) Geef een krappe bovengrens (tight upper bound) op de looptijd (run time) van je algoritme en beargumenteer dit kort.

Antwoord: Dit algoritme loopt in $O(n \log n)$.

3. Gegeven is een verzameling S van punten in het platte vlak. In het algoritme voor het vinden van de twee punten die het dichtst bij elkaar liggen (closest pair of points) worden twee recursieve aanroepen gedaan, links en rechts van een middellijn $x = L$. De kleinste van de twee gevonden afstanden noemen we δ .

(a) (3 punten) Hoe kun je in lineaire tijd bepalen of δ de kleinste afstand is tussen twee punten in S ?

Antwoord: Als voor ieder punt (x_1, y_1) met $x_1 \in [L - \delta, L + \delta]$ de afstand tot ieder van de 11 (of 5 of 6 of...) opvolgende punten (x_2, y_2) met $x_2 \in [L - \delta, L + \delta]$ met grotere y coördinaat groter of gelijk is aan δ .

Leerdoel: 3, D&C.

Niveau: Kennis.

- 1 punt voor bandbreedte $[L - \delta, L + \delta]$
- 2 punten voor vergelijken met constant aantal anderen

(b) (6 punten) Bewijs waarom je antwoord bij (a) altijd een correct antwoord oplevert.

Antwoord: Indien niets vermeld, ken 1 punt toe aan ieder van de volgende onderdelen.

- Punten verder van L dan δ kunnen niet in een paar betrokken zijn met onderlinge afstand kleiner dan δ (omdat ze ofwel beide aan dezelfde kant van L liggen en dan in de recursie gevonden zouden zijn, ofwel aan weerszijde van L liggen en verder van L dan δ). (2 punten)
- (Het is voldoende om ieder punt te vergelijken met punten met een grotere of gelijke y -coördinaat (omdat er voor ieder paar 1 punt is met een grotere (of gelijke) y -coördinaat).)
- Bekijk de twee kolommen van vierkantjes van grootte $\frac{1}{2}\delta$ bij $\frac{1}{2}\delta$ aan weerszijden van L . Er kan maximaal 1 punt per vierkant zijn, omdat ze anders dichterbij elkaar liggen dan δ . (2 punten)
- Er kunnen geen punten van 3 rijen verderop dichterbij (x_1, y_1) liggen dan δ , want de y -coördinaat is dan minstens δ groter. We hoeven dus alleen alle punten binnen 0, 1, of 2 rijen te bekijken. Er zijn maximaal 4 punten te bekijken per rij, dus met 3 rijen maximaal 11 andere punten. (2 punten)

4. (6 punten) Geef een asymptotische boven- en ondergrens voor $T(n)$ in de volgende recurrente betrekking. Maak je grenzen zo krap (tight) mogelijk en geef aan hoe je aan je antwoord komt (in maximaal 5 regels).

$$T(n) = 3T(n/3) + n$$

Antwoord: $\log_b a = \log_3 3 = 1$, dus $f(n) = n$ is $\Theta(n^{\log_3 3})$. Dit is geval 2 van de Master Method. Dus $T(n)$ is $\Theta(n \log n)$.

5. De TU Delft wil graag grote schermen langs de A4 plaatsen om reclame maken voor onderzoeksresultaten. Rijkswaterstaat heeft hiertoe n mogelijke locaties aangegeven. Deze locaties worden geïdentificeerd door hun afstand vanaf Amsterdam, in oplopende volgorde: $x_1, x_2, \dots, x_n$. Uit een voorstudie volgt voor ieder van de locaties een geschat (positief) effect $v_1, v_2, \dots, v_n$. Ook wordt aangenomen dat het totale effect gelijk is aan de som van de effecten op de gekozen locaties. Van Rijkswaterstaat moet er een minimale afstand van 5km tussen de schermen zijn.

(a) (6 punten) Geef een recursieve functie die uitdrukt wat het maximaal haalbare totale effect is. Geef ook aan met welk(e) argument(en) de eerste aanroep gedaan wordt en leg je formule kort uit.

Antwoord: Eerst hebben we een functie nodig die voor iedere locatie i aangeeft wat de verste locatie voor i is die op tenminste 5km afstand staat:

$$p(i) = \max \{ j \in \{0, 1, \dots, n\} \mid j < i \text{ and } x_i - x_j \geq 5 \}$$

waarbij $x_0 \leftarrow -\infty$.

$$\text{OPT}(i) = \begin{cases} 0 & \text{if } i = 0 \\ \max \{ \text{OPT}(i-1), v_i + \text{OPT}(p(i)) \} & \text{otherwise} \end{cases}$$

Roep de functie aan met $\text{OPT}(n)$.

Ook mag $\text{OPT}(1) = v_1$ als basis gebruikt worden, dan is x_0 bij p niet nodig.

- -1 voor vergeten 5km
- -1 voor geen max in $p()$
- +1 voor
- alle punten bij 2 argumenten (maar zie opgave b)
- -2 als max voor de distance check (of ontbreken aanroep $p()$)
- -2 als een van beide in de max ontbreekt
- -1 als v_i op verkeerde plek

- (b) (5 punten) Geef pseudocode voor een efficiënt *iteratief* algoritme om het maximaal haalbare totale effect te berekenen. (Geef aan wat het resultaat is, bv. door een return of print.)

Antwoord:

```

p[0] := 0
j := 0
for i := 1 to n do
    while  $x_i - x_{j+1} \geq 5$  do
        j := j + 1
    end
    p[i] := j
end
M[0] := 0
for i := 1 to n do
    M[i] := max(M[i-1], M[p[i]] + v_i)
end
print M[n]
```

- +2 als iteratief
- +3 voor overeenkomst met a)
- -2 als kwadratisch geheugen (en tijd)
- -0.5 als alleen kwadratisch tijd
- -1 als $p()$ niet geïmplementeerd

6. De Noorse Spoorwegmaatschappij (NS) wil weten of ze voldoende materieel heeft. Het Noorse spoorweginet is opgedeeld in n trajecten die ieder precies een dag duren om af te leggen. Voor ieder traject $i \in \{1, 2, \dots, n\}$ heeft de Noorse regering een minimaal (c_i^-) en maximaal (c_i^+) aantal treinen per dag ingesteld. Een complicerende factor is dat voor ieder traject i slechts bepaalde treintypes j geschikt zijn.

Voor ieder treintype $j \in \{1, 2, \dots, k\}$ is het aantal t_j bekend dat de NS van dit type ter beschikking heeft. De vraag is of dit voldoende is om aan alle eisen te voldoen.

- (a) (5 punten) Modelleer dit probleem door middel van een kringloop met ondergrenzen (circulation with lower bounds). Definiëer alle aspecten van het betreffende model (in maximaal 8 regels). Maak ook een tekening van een eenvoudig voorbeeld met drie trajecten en twee treintypen.

Antwoord: We model this problem by a flow network such that a valid flow represents a valid assignment.

- Introduce a node j for every train type j . (0.5 point)
- Introduce a node i for every trajectory i . (0.5 point)
- Define a source s , and connect s with a directed edge to each j with capacity t_j . (1 point)
- Additionally, define a sink t , and connect each i to t with a directed edge with capacity (upper and lower bounds) $[c_i^-, c_i^+]$. (1 point)
- Connect each node j to all nodes i , if j is suitable for i with capacity ∞ . (1 point)
- Connect the sink t with the source s with a capacity ∞ (optionally with a lower bound $\sum_i c_i^-$) to make the circulation. (1 point)

Note that an equivalent (correct) network is possible where the available supply is connected to the sink, and the demand to the source.

- (b) (4 punten) Leg uit hoe je uit deze kringloop kan afleiden of de NS voldoende materieel ter beschikking heeft (in maximaal 6 regels).

Antwoord: The translation consists of two steps:

1. To get rid of the lower bound c^- on every edge (v, w) increase the demand on v with c^- and reduce the demand on w with c^- . Then reduce the capacity on (v, w) with c^- . (1 point)
2. Create a (new) source s' and connect it with every node which has a negative demand with an edge with capacity equal to the inverse of this demand. Create a (new) sink t' and connect it to every node with a positive demand with an edge with capacity equal to this demand. (2 points)

The NR has enough material if the maximum flow in this network is equal to the total (positive) demand, which is $\sum_i c_i^-$ in this case. (1 point)

- b/c: lower bound not added: -2
- just "do max flow" without discussing lower bound: 1 point
- correct figure may receive all points
- connect all trains with all trajectories: -1
- "mtse" means: missing (t, s) edge: -1

- (c) (2 punten) Leg uit hoe een toewijzing van treinen aan trajecten gevonden kan worden.

Antwoord: By adding the flow on the edges from train types j to trajectories i to the respective lower bound, we get how many trains of type j should be allocated to trajectory i .

7. (10 punten) Gegeven een stroomnetwerk en een geldige stroom f op dit netwerk. Bewijs het volgende.

Als er geen stroomverhogend pad (*augmenting path*) ten opzichte van f bestaat, dan bestaat er een s - t snede (A, B) zodanig dat $v(f) = \text{cap}(A, B)$. (Hint: maak gebruik van het *flow value lemma*.)

Antwoord:

- Laat A de knopen zijn die vanuit s bereikt kunnen worden in de restgraaf, laat B de overige knopen zijn. (3 punten) Dan is (A, B) een s - t snede, want $s \in A$ en er is geen pad van s naar t in de restgraaf, dus $t \notin A$.
- Voor iedere kant (u, v) in de snedeverzameling S (dus $u \in A, v \in B$) geldt dat $f(u, v) = \text{cap}(u, v)$, want anders zou v nog bereikbaar zijn in de restgraaf. (2 punten)
- Ook is $f(v, u) = 0$, om dezelfde reden. (2 punten)
- Met het flow value lemma geldt dat de waarde van de stroom gelijk is aan de netto stroom over iedere snede, dus $v(f) = \sum_{(u,v) \in S} (f(u, v) - f(v, u))$ (Alternatief: $v(f) = \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)$). (2 punten)
- Omdat $f(v, u) = 0$ en $f(u, v) = \text{cap}(u, v)$ geldt dat $v(f) = \text{cap}(A, B)$. (1 punt)

Houd bij de beoordeling rekening met het volgende:

- Indien geen onderscheid tussen kanten (u, v) en (v, u) met $u \in A$ en $v \in B$: -2 punten.
- Bij direct concluderen dat f een max flow is of ander gebruik van de stelling "max flow = min cut": maximaal 4 punten (dus: -6), want het gevraagde bewijs is juist om dat te beargmenteren.
- Bewijs door contrapositie (redenering neemt aan dat alle snedes een flow hebben ongelijk aan de capaciteit): max 10 punten, te bewijzen: als geen flow en snede zodat $v = \text{cap}$ dan is er een stroomverhogend pad (4p). Bewijs: laat stroom gegeven zijn, dan geldt voor alle snedes $f(v) < \text{cap}(A, B)$ (2p). Doe nu een BFS vanaf s in de restgraaf. Dan is t bereikbaar, want anders zou er een snede zijn met $f(v) = \text{cap}(A, B)$. Tegenspraak (3p). Dus er is een stroomverhogend pad. (1p)
- bewijs in andere richting (redenering neemt aan dat er een snede is met flow gelijk aan capaciteit): max 4 punten
- bewijs van weak duality (dat $v(f) \leq \text{cap}(A, B)$): 3 punten

Meerkeuzevragen

1. Gegeven $f(n) = n^2$ en $g(n) = n^2 \log n$. Welke van de volgende beweringen is waar?

- A. $f(n)$ is $O(g(n))$.
- B. $g(n)$ is $\Theta(f(n))$.
- C. $f(n)$ is $\Omega(g(n))$.
- D. $g(n)$ is $O(f(n))$.

Antwoord: A Leerdoel: Looptijd-notatie.
Niveau: Kennis.

2. Laat een ongerichte graaf G met n knopen (nodes) gegeven zijn. Welke van de volgende bewering is **niet** geldig?

- A. Als G tenminste n kanten heeft is G verbonden.
- B. Als G tenminste n kanten heeft, bevat G een cycle.
- C. Als G een boom is dan heeft G ten hoogste $n - 1$ kanten.
- D. Als G verbonden is dan heeft G tenminste $n - 1$ kanten.

Antwoord: A

3. Gegeven een verzameling van n open intervallen (starttijd, eindtijd) in $\mathbb{R}^+$. Hoe zou je de intervallen sorteren om in $\mathcal{O}(n \log n)$ tijd te kunnen bepalen...

- 1. ...wat het grootste aantal overlappende intervallen is (interval partitioning)?
- 2. ...wat het grootste aantal niet-overlappende intervallen is (interval scheduling)?

Sorteer

- A. 1. aflopend op starttijd en 2. aflopend op starttijd
- B. 1. oplopend op starttijd en 2. oplopend op eindtijd
- C. 1. aflopend op eindtijd en 2. oplopend op starttijd
- D. 1. oplopend op eindtijd en 2. oplopend op starttijd

Antwoord: Correct zijn voor 1: oplopend op starttijd of aflopend op eindtijd en voor 2: oplopend op eindtijd of aflopend op starttijd. Dus dat is alleen case B.

4. Voor een tekst is van iedere letter de (unieke) frequentie gegeven. Laat bovendien een optimale prefix codering voor deze tekst gegeven zijn als een binaire boom. Welke bewering is geldig?

- A. De twee laagst frequente letters hebben dezelfde parent (zijn siblings).
- B. De twee laagst frequente letters zitten op het diepste niveau in de boom.
- C. De boom is gebalanceerd.
- D. Ieder van de bovenstaande beweringen is geldig.

Antwoord: A is niet noodzakelijke waar, want het zou kunnen dat er meer dan 2 letters op het diepste niveau in de boom zitten. Echter het algoritme van Huffman produceert dan altijd bomen waarbij dit waar is. Slechte vraag dus. B is altijd waar. C is meestal niet waar.

5. Geef een krappe asymptotische boven- en ondergrens voor $T(n)$ in de volgende recurrente betrekking.
 $T(n) = 6T(n/2) + n^3$

- A. $\Theta(n^2 \log n)$
- B. $\Theta(n^3)$
- C. $\Theta(n^3 \log n)$
- D. $\Theta(n^3 \log^2 n)$

Antwoord: $\log_b a = \log_2 6 = 2.585$, dus $f(n)$ is $\Omega(n^{2.585})$. Dit is geval 1 van de Master Method (meeste werk in wortel). Check regularity condition: voor $6\left(\frac{n}{2}\right)^3 \leq cn^3 \log n$ bestaat een constante $c < 1$, namelijk $c = \frac{6}{2^3} = \frac{3}{4}$. Dus $T(n)$ is $\Theta(n^3)$.

B

Leerdoel: Looptijd.

Niveau: Kennis.

6. Je hebt twee zeer grote, gesorteerde arrays met getallen beide met een lengte $\Theta(n)$. Wat is de juiste asymptotische grens (bound) op de looptijd (running time) van een zo efficient mogelijk algoritme dat de mediaan van de verzameling van de elementen van beide arrays bepaalt?¹

- A. $\Theta(\log n)$
- B. $\Theta(n)$
- C. $\Theta(n \log n)$
- D. $\Theta(n^2)$

Antwoord: A

Leerdoel: 3, D&C.

Niveau: Kennis.

7. Karatsuba heeft een efficiënt recursief (verdeel-en-heers) algoritme bedacht voor het vermenigvuldigen van twee grote gehele getallen. Welke recurrente betrekking beschrijft de looptijd van dit algoritme?

- A. $T(n) = 2T(n/2) + cn$
- B. $T(n) = 3T(n/2) + cn$
- C. $T(n) = 2T(n/3) + cn$
- D. $T(n) = 3T(n/3) + cn$

Antwoord: B

8. Gegeven is een verzameling punten $(x_1, y_1), \dots, (x_n, y_n)$, gesorteerd op x coördinaat. Gegeven is ook voor ieder paar $i < j$ de (minimale) fout $e_{i,j}$ die wordt verkregen als de punten i tot en met j worden benaderd met één lijnstuk.² Verder zijn de kosten C gegeven voor het introduceren van een extra lijnstuk.

In het volgende algoritme voor het berekenen van de minimale kosten voor het benaderen van de punten 1 tot n met behulp van lijnstukken is één regel incompleet.

¹De mediaan van een verzameling is het middelste element van een gesorteerde lijst van alle elementen.

²Hier toe wordt de minimale som van de kwadraten van de Euclidische afstanden van de punten i tot en met j tot het lijnstuk berekend.

```

1 for  $j \leftarrow 1$  to  $n$  do
2    $M[j] \leftarrow e_{1,j} + C$ 
3   for  $i \leftarrow 2$  to  $j$  // als  $j \geq 2$ 
4   do
5      $m \leftarrow \dots$ 
6     if  $m < M[j]$  then
7        $M[j] \leftarrow m$ 
8     end
9   end
10 end
11 print  $M[n]$ 

```

Welke invulling van regel 5 leidt tot het gewenste resultaat?

- A. $m \leftarrow e_{i,j} + C + M[i - 1]$
- B. $m \leftarrow e_{1,j} + C + M[j]$
- C. $m \leftarrow \min\{e_{i,j} + C, M[i - 1]\}$
- D. $m \leftarrow \min\{e_{i,j}, C + M[i]\}$

Antwoord: A

Leerdoel: 7, looptijd.

Niveau: Inzicht.

9. Gegeven zijn n taken en voor ieder van de taken $i \in \{1, \dots, n\}$ een vaste (geheeltallige) duur w_i . We hebben gedurende W tijd op een machine om taken uit te voeren. Het doel is om een deelverzameling S van de taken te kiezen zodat $\sum_{i \in S} w_i \leq W$ en verder zodat $\sum_{i \in S} w_i$ zo groot mogelijk is. Wat is de krapste worst-case bovengrens op de looptijd van de meest efficiënte implementatie om dit probleem op te lossen?

- A. $\mathcal{O}(n)$
- B. $\mathcal{O}(n \log n)$
- C. $\mathcal{O}(n^2)$
- D. $\mathcal{O}(nW)$

Antwoord: D

Leerdoel: 4, DP.

Niveau: Toepassen.

10. Voor je verjaardag heb je P euro verjaardagsgeld gekregen in plaats van de kadootjes op je verlanglijstje. Voor ieder van de n kadootjes ken je de prijs p_i . Je besluit om zoveel mogelijk van je verjaardagsgeld te gebruiken om de kadootjes van je lijstje aan te schaffen. Dit probleem kan efficiënt opgelost worden door dynamic programming op basis van de volgende recursieve functie.

$$\text{OPT}(n, P) = \begin{cases} 0 & \text{als } i = 0, \\ \text{OPT}(i - 1, p) & \text{als } p_i > p \\ \dots & \text{anders} \end{cases}$$

Wat moet er op de puntjes (...) staan om te zorgen dat deze functie beschrijft hoeveel geld je maximaal uit kunt geven aan cadeautjes op je lijstje?

- A. $\max_{1 \leq j \leq i} \{\text{OPT}(j, p - p_j) + p_j\}$
- B. $\max\{\max_{1 \leq j \leq i} [\text{OPT}(j, p - p_j) + p_j], \text{OPT}(i - 1, p)\}$
- C. $\max_{1 \leq j \leq i} \{\text{OPT}(i - j, p - p_j) + \text{OPT}(j, p_j) + p_j\}$
- D. $\max\{\text{OPT}(i - 1, p), \text{OPT}(i - 1, p - p_i) + p_i\}$

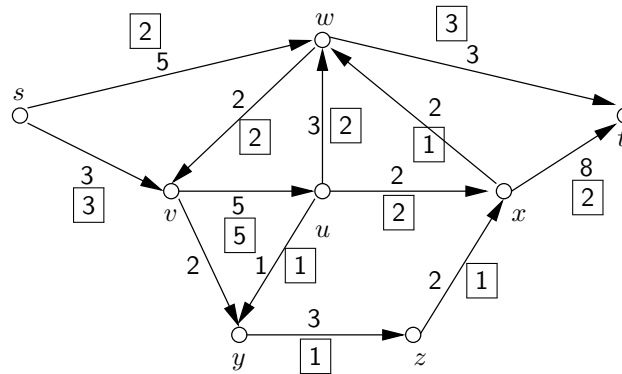
Antwoord: D

11. Het probleem van het vinden van de ruimtelijke structuur van een RNA-molecuul (RNA Secondary Structure) is als volgt: gegeven een rij basen $B = b_1, \dots, b_n$ waarbij iedere $b_i \in \{A, U, C, G\}$, vind een zo groot mogelijke verzameling S van paren (i, j) die aan een aantal voorwaarden voldoen. Als een paar (i, j) aan deze voorwaarden voldoet, schrijven we $p(i, j)$. Hiertoe wordt de volgende functie geïmplementeerd met behulp van dynamisch programmeren.

$$\text{OPT}(i, j) = \max \left\{ \text{OPT}(i, j - 1), \max_{\{t \mid i \leq t < j - 4 \wedge p(t, j)\}} \{1 + \text{OPT}(i, t - 1) + \text{OPT}(t + 1, j - 1)\} \right\}$$

In welke volgorde kunnen de deelproblemen berekend worden?

- A. **for** $i := 1$ **to** $n - 1$ **do**
 for $j := i$ **to** n **do**
 compute $\text{OPT}(i, j)$
 end
 end
- B. **for** $i := n - 1$ **downto** 1 **do**
 for $j := i$ **downto** 1 **do**
 compute $\text{OPT}(i, j)$
 end
 end
- C. **for** $k := 1$ **to** $n - 1$ **do**
 for $i := 1$ **to** $n - k$ **do**
 compute $\text{OPT}(i, i + k)$
 end
 end
- D. **for** $i := 1$ **to** $n - 1$ **do**
 for $k := 1$ **to** $n - i$ **do**
 compute $\text{OPT}(i, i + k)$
 end
 end



Figuur 1: Een stroomnetwerk. De getallen bij de kanten zijn de capaciteiten. De stroom (*flow*) op een kant is weergegeven in een vierkantje.

Antwoord: C

12. Gegeven is een gerichte graaf $G = (V, E)$ met $V = \{u, v, w\}$ en de volgende gerichte kanten met onder- en bovengrenzen.

kant	ondergrens	bovengrens
(u, v)	4	7
(v, w)	5	9
(w, u)	2	10

Beschouw een vertaling van dit probleem naar een netwerkstroom probleem G' zonder ondergrenzen en zonder demands zodat een maximale stroom van G' aangeeft of er een geldige kringloop (*circulation*) bestaat in G . Bestaat er dan een kant (s, v) of (v, t) en wat is de capaciteit c op deze kant?

- A. $c(v, t) = 1$
- B. $c(v, t) = 2$
- C. $c(s, v) = 1$
- D. $c(s, v) = 3$

Antwoord: Het verwijderen van ondergrenzen leidt tot demands $d(u) = 2$, $d(v) = 1$ en $d(w) = -3$. Omdat $d(v) > 0$ wordt die verbonden met t . Dus: A
 Leerdoel: 5, Stroom.
 Niveau: Toepassen.

13. Bekijk het stroomnetwerk in figuur 1. Met hoeveel kan de s - t stroom nog worden verhoogd? (Als het je helpt, teken de rest graaf.)

- A. 0
- B. 1
- C. 2
- D. 3

Antwoord: C

14. Bekijk weer het stroomnetwerk in figuur 1. Wat is de capaciteit van de snede $(\{s, w, v\}, \{u, y, z, x, t\})$?

- A. 2
- B. 5
- C. 8
- D. 10

Antwoord: D
 Leerdoel: 5, Stroom.
 Niveau: Toepassen.

15. Van het algoritme van Ford-Fulkerson bestaat ook een variant die een stuk efficiënter is door het gebruik van een schaling (scaling) techniek. Welk van de volgende beschrijvingen geeft het beste het idee van deze techniek weer?
- A. Zoek in iedere iteratie een stroom van ten hoogste Δ en halveer Δ na iedere iteratie.
 - B. Zoek in iedere iteratie een stroom van tenminste Δ en verdubbel Δ na iedere iteratie.
 - C. Bekijk in iedere iteratie alleen kanten met een restcapaciteit van minimaal Δ en halveer Δ na iedere iteratie.
 - D. Bekijk in iedere iteratie alleen kanten met een restcapaciteit van minimaal Δ en verdubbel Δ na iedere iteratie.

Antwoord: C