

Tentamen TI2306 Algoritmiek

23 januari 2015, 9.00-12.00

- Het gebruik van boek, aantekeningen of rekenmachine tijdens dit tentamen is **niet** toegestaan.
- Dit tentamen heeft 7 open vragen (onderverdeeld in 24 punten) in totaal goed voor 6/10 van je cijfer en 15 meerkeuzevragen goed voor 3/10 van je cijfer. Je krijgt sowieso 1/10 kado. 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).
- 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.
 - 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 4.4, 4.9*, 5.6, 6.8–6.10*, 7.4*, en 7.13*), en bovendien de notities over de Master Method en over bewijstechnieken (Proving guide).
- Totaal aantal pagina's (zonder dit voorblad): 6.

Open vragen

1. (4 punten) Laat een verzameling taken S gegeven zijn die uitgevoerd dienen te worden op één machine. Van iedere taak $j \in S$ is de duur (processing time) p_j gegeven. Het doel is om een volgorde voor het uitvoeren van alle taken S te bepalen waarbij de som van de eindtijden (completion times) $\sum_{j \in S} c_j$ minimaal is. De eindtijd van een taak, c_j , is hierbij gedefinieerd als de som van alle processing times van taken tot en met j .

Bewijs met behulp van een uitwisseling (exchange) argument dat de optimale volgorde om de taken uit te voeren is om ze olopend te sorteren op p_j .

- Geef aan welke bewijstechniek je gebruikt.
- Leg eventuele extra gebruikte notatie uit.
- Maak duidelijk wanneer je een aanname doet.
- Geef een duidelijke verwijzing als je een stelling gebruikt.
- Zorg dat iedere stap in je bewijs volgt uit voorgaande beweringen.

2. In het algoritme voor het vinden van de twee punten die het dichtst bij elkaar liggen (closest pair of points) worden de punten binnen een bandbreedte van δ van een middellijn $x = L$ gesorteerd op y -coördinaat.

(a) (1 punt) Wat is de betekenis van deze δ en hoe wordt die bepaald?

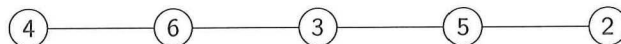
(b) (2 punten) Tot welke andere punten in deze lijst wordt in dit algoritme voor een punt i de afstand bekeken om zo tot het tweetal te komen met de kleinste onderlinge afstand? Leg uit waarom dit een voldoende voorwaarde is.

3. (3 punten) Geef een krappe asymptotische boven- en ondergrens voor $T(n)$ in de volgende recurrente betrekking. Geef aan hoe je aan je antwoord komt (in maximaal 5 regels).

$$T(n) = 9T(n/3) + n^3 \log n$$

4. (3 punten) Er is een padgraaf $G = (V, E)$ gegeven met voor iedere knoop v_i een gewicht w_i . De knopen $v_1, v_2, \dots, v_n$ in een padgraaf zijn genummerd zodat er een kant $(v_i, v_j) \in E$ is dan en slechts dan als $j = i + 1$. We zijn op zoek naar een deelverzameling S van de knopen zodanig dat

- geen van de knopen in S elkaars burens zijn (een zogenaamde *independent set*),
- de som van de gewichten van de knopen in S zo groot mogelijk is.



Figuur 1: Een voorbeeld van een padgraaf met gewichten (de getallen in de knopen).

Definieer het gewicht van een *independent set* van G met de grootste som van gewichten door middel van een recursieve functie. Beschrijf ook met welk argument de functie het antwoord voor de hele graaf geeft. (Hint: denk om de randgevallen en leg je functie kort uit.)

5. Laat $G = (V, E)$ een gerichte graaf zijn met knopen $v_1, v_2, \dots, v_n$ en waar (i) voor iedere gerichte kant (v_i, v_j) geldt dat $i < j$ en (ii) er voor iedere knoop v_i behalve v_n er een knoop v_j is zodat $(v_i, v_j) \in E$. De lengte van het langste pad is recursief gedefinieerd door de volgende functie L aan te roepen met $L(1)$.

$$L(i) = \begin{cases} 0 & \text{if } i = n \\ \max_{(v_i, v_j) \in E} L(j) + 1 & \text{otherwise} \end{cases}$$

- (2 punten) Geef pseudocode voor een efficiënt iteratief dynamisch programmeren algoritme gebaseerd op de functie L (maximaal 10 regels).
- (1 punt) Geef een krappe boven grens (tight upper bound) op de looptijd (run time) van dit algoritme en leg kort uit.

6. (3 punten) Gegeven is een stroom netwerk $G = (V, E)$, met capaciteiten $c : V \rightarrow \mathbb{N}$. Gegeven is een stroom $f : V \rightarrow \mathbb{N}$ op dit netwerk. Bewijs dat voor iedere willekeurige $s - t$ snede (A, B) geldt dat $v(f) \leq \text{cap}(A, B)$.

(Hints: 1. Geef duidelijk aan waar je de definities van $v(f)$ en $\text{cap}(A, B)$ gebruikt. 2. Je mag het flow value lemma gebruiken.)

7. Een aantal studenten heeft een contract met FlexD om wat bij te verdienen. Een contract tussen een student i en FlexD beschrijft het exacte aantal uur (per maand) van de aanstelling en de taken die de student kan uitvoeren. Iedere taak j vereist een zeker aantal uur (per maand).

(We nemen aan dat er geen verschil is tussen 1 student voor 2 uur aanstellen of 2 studenten voor ieder 1 uur en we weten dat het totale aantal uur benodigd voor alle taken gelijk is aan het totaal aantal uur van alle aanstellingen.)

We bekijken het probleem van het toewijzen van studenten aan taken zodat aan alle contracten en taakeisen voldaan wordt.

- (a) (3 punten) Beschrijf een netwerkmodel dat alle elementen van het probleem bevat. Geef ook een schets van een voorbeeld van zo'n netwerk.
- (b) (1 punt) Geef kort aan hoe het algoritme van Ford-Fulkerson op je model uitgevoerd kan worden en hoe je dan kunt zien of er een geldige toewijzing is.
- (c) (1 punt) Beschrijf hoe je een geldige toewijzing van studenten aan taken kunt afleiden (in maximaal 3 regels).

Meerkeuzevragen

1. Voor je volgende project moet je van n taken de looptijd meten. Voor ieder van de taken $i \in \{1, \dots, n\}$ ben je naar verwachting t_i minuten bezig met het implementeren en daarna heeft die taak naar verwachting p_i minuten nodig om uitgevoerd te worden op een van de vele pc's op de Drebbelweg. Dit laatste kan parallel.

In welke volgorde moet je de taken implementeren om zo snel mogelijk met het project klaar te zijn?

- A. Aflopend op t_i
- B. Oplopend op t_i
- C. Aflopend op p_i
- D. Oplopend op p_i

2. Gegeven is de volgende codering voor een taal $S = f, g, h, i$:

$$c(f) = 01$$

$$c(g) = 001$$

$$c(h) = 000$$

$$c(i) = 1$$

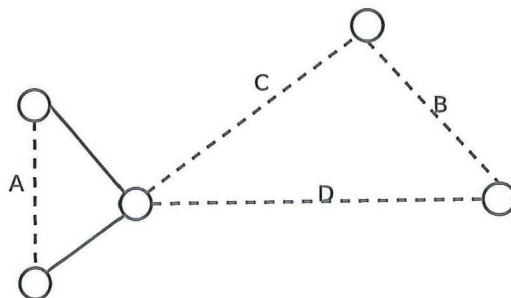
Welk van de volgende stellingen is correct?

- A. De codering c is een geldige prefix code.
- B. De tekst 000010110010 is een encoding van $hfig$.
- C. De tekst 010110011001 is een encoding van $ffifig$.
- D. Iedere geldige encoding eindigt met een 1.

3. Gegeven is een snede (cut) (A, B) in een graaf G met op iedere kant andere kosten. Noem e de kant in de snedeverzameling (cut set) van (A, B) met de minste kosten. Wat kunnen we nu concluderen?

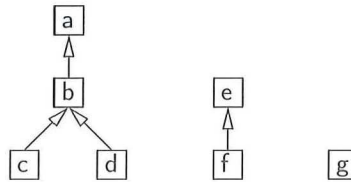
- A. Voor e geldt de cycle property.
- B. Zonder e is de graaf niet verbonden.
- C. Alle minimale opspannende bomen (MSTs) bevatten e .
- D. Er is een kant f in de snedeverzameling die in geen enkele minimale opspannende boom voorkomt.

4. De doorgetrokken kanten (solid edges) in de onderstaande graaf zijn de gekozen kanten (selected edges) tijdens een uitvoering van het algoritme van Prim. De gewichten van een kant zijn evenredig met de afstand tussen de knopen op papier (the edge weights are proportional to their length on paper). Welk van de gestippelde (dashed) kanten zal dit algoritme als eerstvolgende (next) selecteren om toe te voegen aan de gekozen kanten?



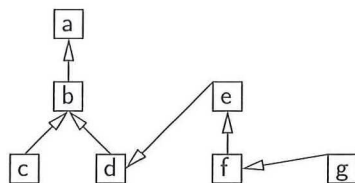
- A. kant A
- B. kant B
- C. kant C
- D. kant D

5. Stel dat we de meest efficiënte pointer-based implementatie van union-find gebruiken *met padcompressie* en op een bepaald moment ziet de datastructuur voor de elementen a t/m g er uit als in figuur 2.

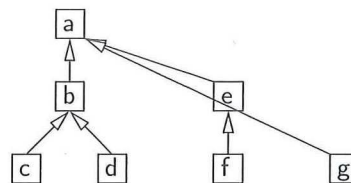


Figuur 2: In deze union-find datastructuur geven de pijlen pointers aan en de vierkanten de data.

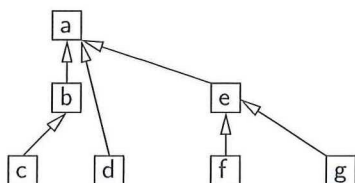
We voeren vervolgens de volgende (zes) operaties uit: $\text{union}(\text{find}(d), \text{find}(f))$ en $\text{union}(\text{find}(g), \text{find}(f))$. Welk van de figuren geeft dan correct het eindresultaat weer?



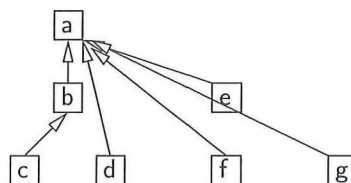
A.



B.



C.



D.

6. Het berekenen van een geheel-tallige (integer) macht (power) $k > 1$ van een getal $x > 0$ kan door middel van verdel- en heers efficiënt gedaan worden. Hoeveel vermenigvuldigingen (multiplications) zijn er nodig om x^{81} door te rekenen?
- A. 6
 - B. 8
 - C. 9
 - D. 40
7. Wat is de krapste bovengrens op de looptijd van het meest efficiënte algoritme voor het vinden van het paar dichtstbijzijnde punten in het platte vlak (closest-pair-of-points)?
- A. $O(n)$
 - B. $O(n \log n)$
 - C. $O(n \log^2 n)$
 - D. $O(n^2)$

8. Geef een zo krap (tight) mogelijke asymptotische bovengrens voor de volgende recurrente betrekking:
 $T(n) = 2T(n/2) + n\sqrt{n}$.

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

9. 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.¹ 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.

```

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. $m \leftarrow \min\{e_{i,j}, C + M[i]\}$

10. Laat n taken gegeven zijn met voor iedere taak j een waarde v_j , en een (vaste) begin en eindtijd, respectievelijk s_j en f_j . Geef een recursieve functie voor het bepalen van de grootste waarde van een deelverzameling van niet-overlappende (non-intersecting) taken. Je mag gebruik maken van een predecessor functie $p(j)$ die voor taak j de niet-overlappende taak geeft voorafgaand aan j met de laatst mogelijke eindtijd. Gegeven is ook $\text{OPT}(0) = 0$ en verder:

- A. $\text{OPT}(j) = \max_{1 \leq i < j} \{v_i + \text{OPT}(p(i))\}$
- B. $\text{OPT}(j) = \max_{1 \leq i < j} \{v_j + \text{OPT}(p(i))\}$
- C. $\text{OPT}(j) = \max \{v_j + \text{OPT}(p(j)), \text{OPT}(j - 1)\}$
- D. $\text{OPT}(j) = \max \{v_j + \text{OPT}(j - 1), \text{OPT}(p(j - 1))\}$

11. 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.

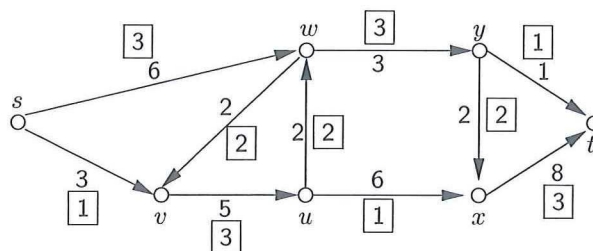
Dit probleem kan efficiënt opgelost worden door dynamic programming. Met welke van de volgende recursieve functies beschrijft $\text{OPT}(n, P)$ hoeveel geld je maximaal uit kunt geven aan cadeautjes op je lijstje?

Als $i = 0$, dan $\text{OPT}(i, p) = 0$, (anders) als $p_i > p$, dan $\text{OPT}(i, p) = \text{OPT}(i - 1, p)$, en anders

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

¹Hiertoe wordt de minimale som van de kwadraten van de Euclidische afstanden van de punten i tot en met j tot het lijnstuk berekend.

12. Gegeven is een graaf (V, E) met twee bijzondere knopen genaamd $s \in V$ en $t \in V$ met $s \neq t$. Hoe kun je efficient bepalen hoeveel paden er van s naar t bestaan die niet gebruik maken van dezelfde kanten (paths with disjoint edges)?
- Verdeel de graaf herhaaldelijk in tweeën zodanig dat er in ieder deel een pad bestaat.
 - Bereken herhaaldelijk het kortste pad met (bijvoorbeeld) Dijkstra en verwijder dit uit de graaf.
 - Ken aan iedere kant een capaciteit van 1 toe en bepaal dan de maximale stroom van s naar t .
 - Gebruik Floyd-Warshall om alle kortste paden te berekenen en trek de overlappende paden hiervan af.
13. Gegeven is een kant (u, v) met $d(u)=5$ and $d(v) = -3$. De capaciteit van u naar v is 6 en heeft een ondergrens (lower bound) van 2. Neem aan dat de ondergrens op alle andere kanten 0 is.
- Het algoritme van Ford-Fulkerson wordt uitgevoerd op een vertaling (translation) van dit probleem naar een netwerkstroom probleem zonder ondergrens en zonder demand d . De maximale stroom over (u, v) is daarin 3. Wat is de gevonden stroom over (u, v) in het oorspronkelijke netwerk?
- 2
 - 3
 - 4
 - 5
14. Bekijk het stroomnetwerk in figuur 3. Met hoeveel kan de stroom van s naar t nog worden verhoogd? (Als het je helpt, teken de restgraaf.)
- 2
 - 3
 - 4
 - 5
15. Bekijk weer het stroomnetwerk in figuur 3. Wat is de capaciteit van de snede $(\{s, u, v, x\}, \{w, y, t\})$?
- 8
 - 12
 - 14
 - 16



Figuur 3: Een stroomnetwerk. De getallen bij de kanten zijn de capaciteiten. De stroom (flow) op een kant is weergegeven in een vierkantje (square).