

Antwoorden

1: (a) ja, (b) ja, (c) ja, (d) ja, (e) nee.

Toelichting: (a) Sommige hebben hier geantwoord: 'Nee, want $P \subset NP$ '. Er geldt wel dat het vermoeden is dat P een stricte deelverzameling is van NP , maar ook dan zou nog steeds gelden dat $P \subseteq NP$.

(b) Zie complexity notes.

(c) Zie complexity notes.

(d) Het zit in P dus zeker in NP .

(e) Is op het hoorcollege genoemd als voorbeeld dat exponentiele tijd niet hetzelfde is als NP -hard. Het aantal stappen is hier wel exponentieel maar dit is geen optimaliseringsprobleem. (Als je dit als optimaliseringsprobleem zou schrijven dan wordt het: Wat is het kleinste aantal zetten om alles van A naar C te krijgen? Maar dat is bekend: het antwoord is $2^n - 1$ en dit 'optimaliseringsprobleem' is dus in constante tijd op te lossen.

2: *Kruskal*: De lijnen worden eerst gesorteerd op oplopende volgorde van gewicht en daarna in deze volgorde toegevoegd. Indien er een cykel zou ontstaan door toevoeging dan wordt de lijn overgeslagen.

Prim: De boom wordt, beginnende bij een willekeurig punt steeds met 1 punt uitgebreid. Bij de uitbreiding wordt steeds gekozen voor de lijn (u, w) met het kleinste gewicht waarbij u in de huidige boom zit en w niet.

Boruvka: Net als Prim, maar dan wordt dit parallel gedaan. In het begin zijn alle n punten een boom op 1 punt. In een iteratie wordt elke boom uitgebreid met de aangrenzende lijn van minimaal gewicht. (Als alle gewichten verschillend zijn dan zullen er geen cycli ontstaan.) Het algoritme stopt als er nog maar 1 boom is.

3 In het optimale schedule staan de jobs op aflopende volgorde van w_j/p_j . Hier: $\frac{w_3}{p_3} < \frac{w_2}{p_2} < \frac{w_1}{p_1} < \frac{w_4}{p_4}$. De volgorde is dus 3, 2, 1, 4,.

$$C_3 = 2, C_2 = (2 + 5) = 7, C_1 = (2 + 5 + 7) = 14, C_4 = (2 + 5 + 7 + 4) = 18.$$

$$\text{De waarde is } \sum_j w_j C_j = 6 \cdot 2 + 11 \cdot 7 + 9 \cdot 14 + 5 \cdot 18 = 12 + 77 + 126 + 90 = 305.$$

4: (Zie opgaven week 1) (a) Bekijk de som van de graden. Elke lijn wordt dan twee maal geteld: $\sum_{v \in V} d(v) = 2|E|$. De som is dus even. Er is dus een even aantal punten van oneven graad.

(b) Stel dat er geen pad loopt. Dan zitten de twee punten elk in een component met precies 1 punt van oneven graad. Uit (a) volgt dat dat niet kan.

5: Stel T is een MST maar bevat niet lijn e_1 . Als we e_1 aan T toevoegen dan ontstaat er een cykel. Als we een lijn $e \neq e_1$ uit de cykel weglaten dan hebben we weer een opspannende boom T' . Het gewicht van de boom is afgenomen met $w(e) - w(e_1) > 0$. Maar we hadden aangenomen dat T een opspannende boom met minimaal gewicht was. Dit is dus niet mogelijk. Conclusie: elke MST moet e_1 bevatten. (Zie ook opgaven week 1)

6: Voor de DP wordt $F(i)$ berekend in de volgorde $i = 1, 2, \dots, n$.

$F(i)$ is TRUE als $s[1 \dots i]$ een woord is, óf als er een $j < i$ is zodat $F(j) = \text{TRUE}$ en $w = s[j+1, i]$ een woord is ($DICT(w) = \text{TRUE}$).

Elke berekening van $F(i)$ kan dus in $O(n)$ gedaan worden door te kijken naar $F(j)$ voor alle $j < i$.

De eindoplossing wordt gegeven door $F(n)$.

NB: Bovenstaande is dus voldoende als antwoord. Het bevat: (i) de recurrente betrekking, (ii) een analyse van de running time, en (iii) het eindantwoord, $F(n)$ in dit geval.

7: Zie opgaven max flow.