

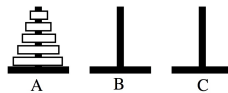
Tussentoets combinatorische optimalisatie 2013-2014.

Tijd: 11.00-12.45. Toets met 'gesloten boek'. Beschrijf bij elke opgave steeds het belangrijkste idee. Een tekening kan soms handig zijn. Notatie en exacte formulering is van minder belang, tenzij anders vermeld. Maximaal zijn 50 punten te behalen.

Opgave 1 (5punten).

Beantwoord de volgende vragen alleen met ja, nee of laat open. Goed antwoord: +1 punt; Fout antwoord: -1 punt; Geen antwoord: +0 punt. (Minimaal aantal punten is 0 punten.)

- (a) $P \subseteq NP$.
- (b) Reducties zijn transitief, dwz., als Π_1, Π_2, Π_3 beslissingsproblemen zijn en Π_1 reduceert tot Π_2 en Π_2 reduceert tot Π_3 dan geldt dat Π_1 reduceert tot Π_3 .
- (c) TSP is *sterk* NP-moeilijk (strongly NP-hard) .
- (d) Het kortse pad probleem (de beslissingsvariant) zit in NP.
- (e) Het Towers of Hanoi probleem (Verplaats alle schijven van pin A naar pin C met een zo min mogelijk aantal zetten.) is een NP-moeilijk (NP-hard) probleem.



Opgave 2 (10 punten)

Beschrijf in woorden de drie verschillende MST algoritmes: Kruskal, Prim, en Boruvka. Een formele beschrijving wordt niet gevraagd maar geef voor elk in een paar zinnen het belangrijkste idee van het algoritme.

Opgave 3 (5 punten)

Beschouw onderstaande instantie van het probleem $1||\sum_j w_j C_j$. Geef een optimaal schedule (=volgorde van jobs) en z'n waarde.

jobs	1	2	3	4
w_j	9	11	6	5
p_j	7	5	2	4

Opgave 4 (5 punten)

- (a) Toon aan dat elke graaf een even aantal punten heeft van oneven graad.
- (b) Toon aan dat als een graaf precies 2 punten van oneven graad heeft, dan loopt er een pad tussen deze punten.

Opgave 5 (10 punten)

Stel $G = (V, E)$ is een simpele gewogen graaf. (Een graaf is simpel als het geen multiple edges of loops bevat.) Verder is G ongericht en verbonden en heeft elke lijn $e \in E$ een gewicht $w(e) \geq 0$. De lijnen van G zijn gelabeld zodat $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$. Stel $w(e_1) < w(e_2)$. Laat zien dat elke Minimum Spanning Tree van G lijn e_1 bevat.

Opgave 6 (10 punten)

Stel $s[1 \dots n]$ is een string van n letters. Je vermoedt dat het eigenlijk een stuk tekst is waarvan de interpunctie (komma's, spaties, punten e.d.) verdwenen is (het lijkt op iets als Jevermoedtdathet...). Je wilt de tekst herstellen met behulp van een woordenboek, welke je kunt raadplegen in de vorm van een Boolean function $\text{DICT}(w)$. Voor een substring w , $\text{DICT}(w)$ is TRUE als w een echt woord is en anders is $\text{DICT}(w)$ FALSE.

Geef een Dynamic Programming algoritme dat bepaalt of een string $s[1 \dots n]$ een aaneenschakeling is van echte woorden. De running time moet $O(n^2)$ zijn, aangenomen dat elke aanroep van de functie $\text{DICT}(w)$ één tijdseenheid kost.

Hint: Definieer een boolean functie $F(i)$ die TRUE is als de prefix $s[1 \dots i]$ (dwz, de eerste i letters van de gegeven string) een aaneenschakeling is van echte woorden. Hoe volgt de waarde van $F(i)$ uit de waarden $F(j)$ voor $j < i$?

Opgave 7 (5 punten) Beschrijf een manier om elk van de volgende varianten van het max flow probleem te reduceren naar het standaard max-flow probleem (dwz, een gericht netwerk met source s en sink t en capaciteiten op de lijnen). Een schets met korte toelichting is voldoende.

- (a) Er zijn meerdere sources: $s_1, s_2, \dots, s_k$ en meerdere sinks $t_1, t_2, \dots, t_k$.
- (b) Het netwerk is ongericht.
- (c) Zowel de lijnen als de punten hebben capaciteiten. (Als een punt v een capaciteit $b(v)$ heeft dan kan er niet meer dan $b(v)$ dat door het punt stromen.)

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.