

Antwoorden

1 (a) Ja (b) nee (c) nee (d) ja (e) ja.

Toelichting: (a) Per definitie van NP -completeitheid is elk probleem reduceerbaar tot Π . Als $\Pi \in P$ dan is elk probleem in NP dus op te lossen in polynomiale tijd door het te reduceren naar Π .

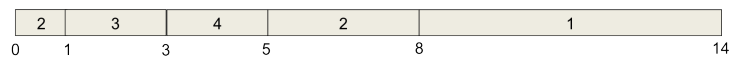
(b) Andersom: Als Π_1 NP -compleet is en Π_1 reduceert tot Π_2 dan is Π_2 ook NP -compleet.

(c) Nee, knapsack (en integer knapsack) is op te lossen in pseudo polynomiale tijd door middel van dynamic programming. (Zie slides en complexity notes.)

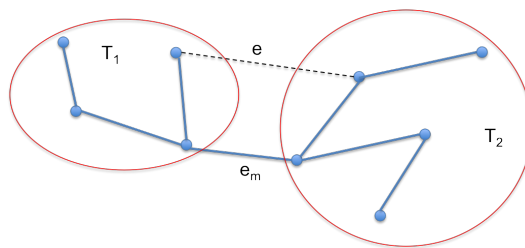
(d) Ja, elk probleem in NP is reduceerbaar tot een NP -moeilijk probleem.

(e) Ja, het Hamiltonian Cycle probleem is reduceerbaar tot het TSP probleem met afstanden 1 en 2. Zie Stelling 1, complexity notes.

2 Het optimale algoritme scheduled altijd in volgorde van Shortest Remaining Processing Time (SRPT). Dat geeft hier onderstaand schedule. De waarde is $C_1 + C_2 + C_3 + C_4 = 14 + 8 + 3 + 5 = 30$.



3 Stel dat we een MST T hebben die e_m wel bevat. We laten zien dat dit niet mogelijk is. Laat e_m weg uit T . De boom valt dan uiteen in twee delen, zeg T_1 en T_2 . Omdat G verbonden blijft als e_m wordt wegelaten moet er dus nog een lijn $e \in E$ zijn die T_1 met T_2 verbindt. Voeg nu e toe na het weglaten van e_m . We krijgen dan weer een boom en het totale gewicht is afgenomen met $w(e_m) - w(e) > 0$. Dit is niet mogelijk want we hadden aangenomen dat T een minimale opspannende boom is.



4 Een bedrag v kan exact betaald worden als er een muntje i is zodat het bedrag $v - x_i$ exact betaald kan worden.

Neem $F(0) = 1$. Er geldt nu dat $F(v) = 1$ als er een $i \in \{1, \dots, n\}$ is waarvoor $F(v - x_i) = 1$. Om $F(v)$ te berekenen hoeven we dus maar n waarden op te zoeken in de tabel. In de tabel staan alle waarden $F(v)$ voor $v = 0, \dots, W$ en elke kan in $O(n)$ tijd berekend worden. Het eindantwoord is $F(W)$.

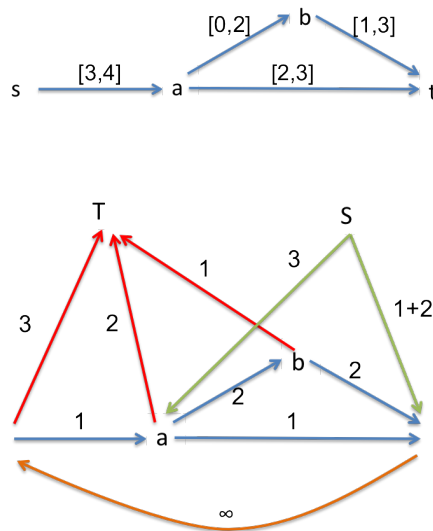
5 Als algoritme nemen we gewoon een minimum spanning tree.

$$\text{OPT}(MST) \leq \text{OPT}(TSP) \leq 2\text{OPT}(ST).$$

Bekijk een optimale TSP en laat een lijn weg. We krijgen dan een opspannende boom: $\text{OPT}(MST) \leq \text{OPT}(TSP)$.

Bekijk een optimale Steiner tree en verdubbel alle lijnen in de oplossing. Maak een Eulertour en pas vervolgens shortcuts toe zodat een TSP tour verkregen wordt: $\text{OPT}(TSP) \leq 2\text{OPT}(ST)$.

6



De ondergrenzen worden vervangen door 0 en van de bovengrens wordt de ondergrens afgehaald. Van s naar t komt een arc met oneindige capaciteit en er worden nieuwe S en T toegevoegd. Voor elke pijl a met ondergrens $b(a)$ gaat er een nieuwe pijl met capaciteit $b(a)$ naar T en een pijl met capaciteit $b(a)$ vanuit S . Er bestaat een toegelaten stroom in het originele netwerk, dan en slechts dan, als er een stroom van S naar T met waarde 6 bestaat in het nieuwe netwerk.

7 Er geldt

$$P(i, j) = p_i P(i-1, j-1) + (1-p_i) P(i-1, j).$$

In de tabel zetten we de waarden $P(i, j)$ voor alle $i \in \{1, \dots, n\}$ en $j \in \{0, \dots, k\}$. Dit zijn $O(n^2)$ waarden en het uitrekenen van een enkele waarde gaat in $O(1)$ tijd. Het antwoord wordt gegeven door $P(n, k)$.

8 Algoritme 1: Neem een MST en verdubbel alle lijnen. Dit geeft een Euler tour. Pas vervolgens shortcuts toe om een TSP tour te krijgen.

Algoritme 2: Bijvoorbeeld Nearest Addition: Start met een enkel punt en breid de tour T steeds verder uit. Kies steeds een punt k die niet in T zit en waarvoor de afstand tot T minimaal is. Stel j is het punt op T met minimale afstand en stel i is buur van j op T . Voeg de lijnen (j, k) en (i, k) toe en laat (i, j) weg. Herhaal dit totdat alle punten op de tour liggen.