

Universiteit Utrecht
Betafaculteit

Examen Discrete Wiskunde II op donderdag 6 juli 2017, 13.30-16.30 uur.

- De opgaven dienen duidelijk uitgewerkt te zijn en netjes ingeleverd te worden. Schrijf op elk ingeleverd vel uw naam en studentnummer.
- Een rekenmachine is niet nodig en derhalve verboden. Eventuele rekenfouten worden niet bestraft, tenzij het de spuigaten uitloopt (dit ter beoordeling van de nakijker).
- Het examen omvat 7 opgaven met in totaal 16 (deel)opgaven.
- Bij de opgaven hieronder **behalve Opgave 3** mag u er van uitgaan dat u een algoritme voor Max Flow, Min cost max flow, Kortste pad, Minimale Opspannende boom beschikbaar hebt (zie lijst). Wanneer u dit wilt toepassen hoeft u alleen aan te geven op welke instantie dit moet gebeuren.
- De maximale score per onderdeel bedraagt:
 - 2 punten voor ieder van de vragen 1b-g, 2a-b;
 - 3 punten voor onderdeel 1a;
 - 4 punten voor ieder van de vragen 2c-d, 3, 4;
 - 5 punten voor ieder van de vragen 5, 6, 7.

Totaal 50 punten.

Succes!

=====

Opgave 1.

Gegeven is een ketting met $k = 6$ kralen; deze kralen kunnen drie kleuren krijgen: rood, blauw en wit. Op deze ketting kun je twee soorten permutaties uitvoeren: wel/niet omdraaien en met 0, 1, 2, 3, 4, 5 plaatsen verschuiven.

- (a) Geef de verschillende permutaties (denk eraan dat je ze bij (b) ook nog nodig hebt).
- (b) Toon aan dat de permutaties van (a) een permutatiegroep vormen; associativiteit hoeft je niet aan te tonen.
- (c) Gegeven de gewichten $w(\text{rood}) = r$, $w(\text{blauw}) = b$ en $w(\text{wit}) = w$, bepaal de pattern inventory (je mag uiteraard machten van $(b + r + w)$ laten staan).

Gegeven de pattern inventory, definieer $Inve(w(\text{rood}), w(\text{blauw}), w(\text{wit}))$ als de waarde van de pattern inventory wanneer je getal $w(\text{rood})$, enz. invoert (bijv. $Inve(2, 2, 2)$ geeft de waarde aan wanneer iedere kleur waarde 2 krijgt). Gebruik deze notatie hieronder. **Je moet bij (d), (e), (f) aangeven hoe je het gevraagde aantal equivalentieklassen kunt bepalen; het antwoord zelf hoeft je niet uit te rekenen.** Gebruik zoveel mogelijk getallen (en dus zo min mogelijk symbolen). **Motiveer uw antwoord.**

- (d) Bepaal het aantal equivalentieklassen zonder witte kralen.
- (e) Bepaal het aantal equivalentieklassen met minstens één rode en minstens één witte kraal.
- (f) Bepaal het aantal equivalentieklassen met een even aantal blauwe kralen.
- (g) Bepaal het aantal equivalentieklassen met een oneven aantal blauwe kralen en meer rode dan witte kralen.

Opgave 2.

Er is werk aan de winkel. In dit geval zijn er n taken die moeten worden uitgevoerd door twee personen: Koos en Toos. Zowel Koos als Toos hebben hun specialiteit, en de taken die daaronder vallen moeten door de betreffende persoon worden gedaan; de overige taken willen ze zo verdelen dat het verschil in hun totale werktijd zo klein mogelijk is (uiteraard in absolute waarde). Noem dit probleem WERK VERDELEN. Van iedere taak T_i ($i = 1, \dots, n$) is de bewerkingduur p_i (neem aan dat dit een niet-negatief geheel getal is) bekend en door wie deze mag worden uitgevoerd (Koos, Toos, of beide).

- (a) Formuleer de beslissingsvariant van het probleem WERK VERDELEN.
- (b) Laat zien dat de beslissingsvariant van het probleem WERK VERDELEN tot de klasse \mathcal{NP} behoort. U mag er hierbij van uitgaan dat de inputgrootte gelijk is aan $O(n \log p_{\max})$, waarbij p_{\max} de grootste bewerkingduur uit de instantie is.
- (c) Toon aan dat het de beslissingsvariant van het probleem WERK VERDELEN \mathcal{NP} -volledig is. U moet hierbij gebruiken dat het probleem PARTITIE \mathcal{NP} -volledig is. Formuleer de reductie en bewijs de correctheid ervan; u hoeft niet te bewijzen dat de reductie polynomiaal is. Het probleem PARTITIE is als volgt gedefinieerd: gegeven t niet-negatieve gehele getallen a_1, \dots, a_t , bestaat er een deelverzameling S van de indexverzameling $\{1, \dots, t\}$ waarvoor geldt

$$\sum_{j \in S} a_j = \left(\sum_{j=1}^t a_j \right) / 2?$$

- (d) Stel dat iemand beweert dat hij een algoritme heeft ontwikkeld dat **voor iedere instantie** van het probleem WERK VERDELEN in polynomiale tijd een toegelaten oplossing vindt die maximaal 10 eenheden boven het optimum zit (dus als het optimum z^* is, dan vindt dit algoritme een oplossing met waarde $\leq z^* + 10$). Welke conclusie kunt u dan trekken? Motiveer uw antwoord.

Opgave 3.

Beschouw het hieronder gegeven netwerk. De getallen bij de pijlen geven respectievelijk de waarde van de huidige stroom door die pijl en de capaciteit van die pijl aan. Bepaal de maximale stroom door dit netwerk en bewijs de maximaliteit van deze stroom met behulp van een minimale snede. **Geef alle stappen van uw berekening duidelijk aan, inclusief residuele graaf. Het uitsluitend vermelden van de oplossing wordt niet goedgekeurd.**

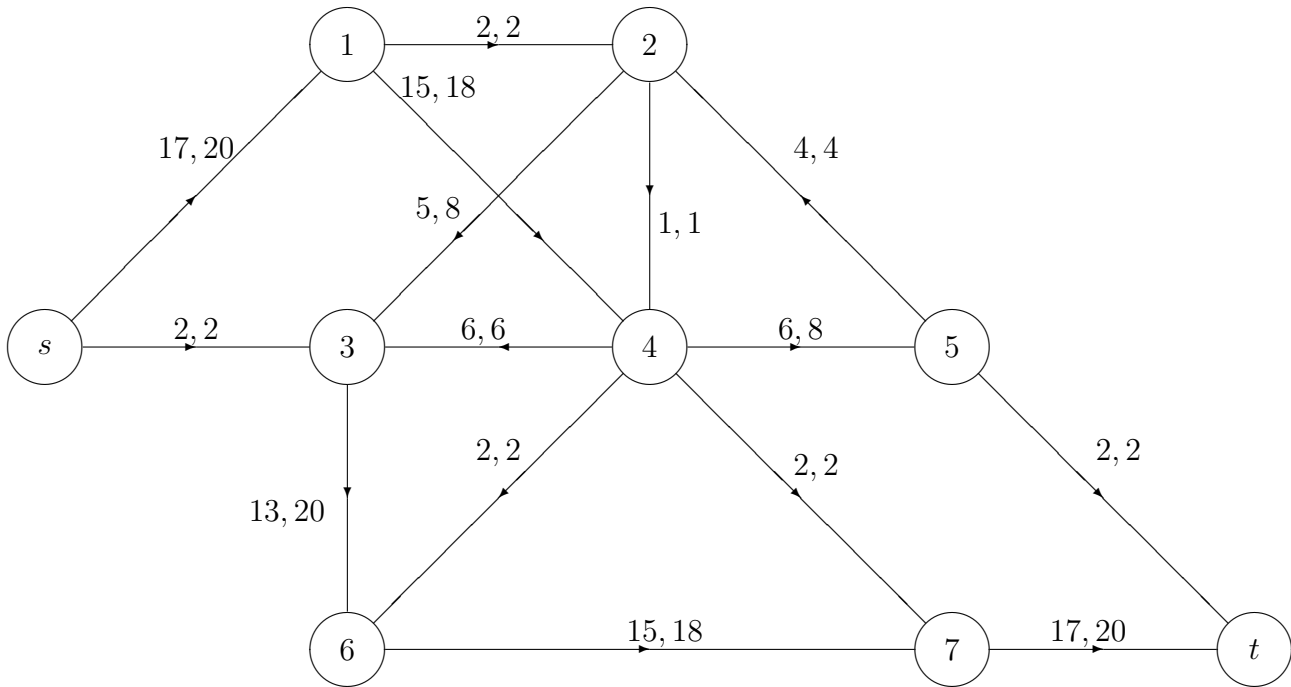


Figure 1: Netwerk

Opgave 4.

Gegeven is een drukbezette docent, die aan m groepen les moet geven. Voor iedere groep i is het aantal lessen r_i ($i = 1, \dots, m$) bekend dat per week gegeven moet worden; iedere les kost 1 uur. De week is verdeeld in p uren, waarbij voor ieder uur bekend is welke groepen dan les kunnen krijgen (de groepen hebben nog meer te doen en zijn dus niet altijd beschikbaar; de docent is altijd beschikbaar). Het doel is een schema te vinden waarin zoveel mogelijk lessen worden gegeven per week. Geef aan hoe je dit probleem op kunt lossen met behulp van één van de op het college behandelde algoritmen (zie onder). **Motiveer uw antwoord.** Geef hierbij ook aan hoe je de oplossing van dat probleem kunt vertalen naar een oplossing voor het roosteringsprobleem.

Opgave 5.

Bij een (voetbal)wedstrijd heb je een arbiter nodig, en bij gebrek aan goede scheidsrechters wil je de aanwezige scheidsrechters natuurlijk zo veel mogelijk koesteren. Scheidsrechter X is zo'n vrijwilliger die iedere zaterdag (en of zondag) druk bezig is om een aantal wedstrijden in goede banen te leiden tegen een marginale vergoeding.

Gegeven is een verzameling van n wedstrijden in de regio, die X zou kunnen fluiten. Van iedere wedstrijd is bekend waar deze wordt gespeeld, wanneer hij begint en wanneer hij eindigt. Uiteraard wil X graag fluiten, maar sommige wedstrijden net iets grager dan andere wedstrijden; neem aan dat je voor iedere wedstrijd j ($j = 1, \dots, n$) weet wat het gewicht w_j (> 0) is dat X aan deze wedstrijd toekent. Na afloop van de wedstrijd krijgt X altijd een kopje koffie aangeboden (duurt 15 minuten) en gaat dan een volgende wedstrijd fluiten op hetzelfde sportcomplex, of hij vertrekt om ergens anders een wedstrijd te gaan fluiten. Neem aan dat je de bijbehorende reistijd $c_{i,j}$ om van de plaats van wedstrijd i naar de plaats van wedstrijd j te komen kent; hier zit ook de tijd in die X nodig heeft om zich voor te bereiden op wedstrijd j . Hoewel X graag weer even ergens gaat kijken, vindt hij het toch vervelend om naar een andere plaats te gaan, wat resulteert in een score y_{ij}

(< 0) voor de verplaatsing. Het doel van X is om het totaal gewicht van de wedstrijden waar hij aan wordt toegewezen (de w_j component) plus de verplaatsingswaardering (de y_{ij} component) te maximaliseren.

Geef aan hoe je dit probleem op kunt lossen met behulp van één van de op het college behandelde algoritmen (zie onder). **Motiveer uw antwoord.** Geef hierbij ook aan hoe je de oplossing van dat probleem kunt vertalen naar een oplossing voor het scheidsrechter-probleem.

Opgave 6.

Gegeven is een gerichte graaf $G = (V, A)$, waarbij $l(v, w)$ de lengte aangeeft van iedere pijl $(v, w) \in A$; deze lengtes zijn allemaal ≥ 0 . De pijlen in dit netwerk moeten worden gecheckt op fouten. Het checken gebeurt door middel van een robot; deze ‘wandelt’ van v naar w (als (v, w) een pijl in de graaf is) en vervolgt dan zijn weg door een andere pijl, of wordt opgehaald door een werknemer. De kosten van de robot bedragen $10l(v, w)$ indien de pijl (v, w) wordt genspecteerd en $l(v, w)$ indien deze pijl alleen wordt doorlopen. De robot mag in een willekeurig punt beginnen; het beginnen in punt $v \in V$ kost b_v ; de robot kan uit ieder willekeurig punt worden opgehaald, maar dit kost een bedrag van Q per keer. Wanneer je de robot uit punt l ophaalt en vervolgens weer laat beginnen in punt v , dan kost dit in totaal $Q + b_v$.

Geef aan hoe je dit probleem op kunt lossen met behulp van één van de op het college behandelde algoritmen (zie onder). **Motiveer uw antwoord.** Geef hierbij ook aan hoe je de oplossing van dat probleem kunt vertalen naar een oplossing voor het inspectieprobleem.

Opgave 7.

Gegeven zijn n mensen die op een punt p aankomen en vervolgens moeten worden getransporteerd naar punt q . Het vervoer gebeurt met bussen (met onbegrensde capaciteit); het inzetten van een bus kost in totaal K , ongeacht het aantal passagiers. Het tijdstip waarop persoon i aankomt is t_i ($i = 1, \dots, n$); dit is van tevoren bekend. Op ieder moment kun je beslissen een bus in te zetten voor de personen die op dat moment aanwezig zijn, of te wachten tot de volgende persoon aankomt. De totale kosten van een oplossing zijn gelijk aan de kosten van het inzetten van de bussen plus de som van de vertrektijden van de n mensen vanuit punt p .

Formuleer een algoritme om dit probleem op te lossen. **Motiveer uw antwoord.** Geef ook aan hoe je de oplossing van dat probleem kunt vertalen naar een oplossing voor het busprobleem. **Hint: begin achteraan.**

=====

Op het college behandelde algoritmen

- Algoritme van Dijkstra: Dit bepaalt het kortste pad van punt s naar ieder ander punt t in een gerichte graaf $G = (V, A)$. Voorwaarde is dat alle afstanden ≥ 0 zijn.
- Algoritme van Bellman-Ford: Als boven, maar het werkt ook als er pijlen zijn met negatieve lengte, zolang er maar geen cykels zijn met negatieve lengte.
- Algoritme van Kruskal: Gegeven een samenhangende, ongerichte graaf $G = (V, E)$ bepaal je een Minimum Spanning Tree door de kanten aan een (oorspronkelijk lege) kantenverzameling S toe te voegen in volgorde van lengte, waarbij een kant weer wordt verwijderd indien deze tot een cykel in de kantenverzameling S leidt.
- Algoritme van Ford-Fulkerson: Dit bepaalt een stroom van maximale omvang van source (bron) s naar sink (put) t in een gerichte graaf $G = (V, A)$, waarbij voor iedere pijl $(v, w) \in A$ een capaciteit $c(v, w)$ is gegeven. Er geldt dat de ondergrens op de stroom door (v, w) gelijk is aan 0.
- Algoritme voor oplossen van min-cost max-flow probleem: Dit bepaalt de goedkoopste stroom van maximale omvang van source (bron) s naar sink (put) t in een gerichte graaf $G = (V, A)$, waarbij voor iedere pijl $(v, w) \in A$ een capaciteit $c(v, w)$ en kosten $k(v, w)$ zijn gegeven. De kosten van een stroom zijn gelijk aan

$$\sum_{(v,w) \in A} k(v, w)x_{vw},$$

waarbij x_{vw} de omvang van de stroom door de pijl (v, w) is. Er geldt weer dat de ondergrens op de stroom door (v, w) gelijk is aan 0.