

Discrete Wiskunde Tentamen 25-05-09

Vergeet het aparte multiple choice gedeelte niet.

1. Dominostenen

Zij $k > 0$ geheel. Op een k -dominosteen staan twee al dan niet verschillende getallen $0, 1, \dots, k$. Een volledige set k -dominostenen bevat elke herhalingscombinatie van twee getallen precies één keer. Het is de bedoeling stenen in een kring te leggen, zó dat op de aangrenzende velden hetzelfde getal staat.

(a) Algemene kennis.

(1) Geef een grafentheoretische beschrijving van een set k -dominostenen.

Neem de volledige graaf op $k+1$ punten $0, 1, \dots, k$ met een lus in elk punt. Een lijn ij met $i \neq j$ stelt de steen voor met de getallen i en j . Een lus in i stelt de steen voor met tweemaal het getal i .

(2) Formuleer de *stelling van Euler* over het bestaan van bepaalde gesloten wandelingen in een graaf G .

Er is een Euler toer enkel en alleen als de graaf evengraads en samenhangend is.

(b) Bij welke waarden van k kun je een kring van k -dominostenen leggen die alle stenen gebruikt, en voor welke kan dat niet? Leg uit.

Voor even k kan je alle stenen opgebruiken, voor oneven k niet. De lussen zijn daarbij niet relevant: de dubbelsteen met getal i kan later in de kring worden gelegd op een plaats waar twee stenen met verschillende getallen aan elkaar gelegd zijn met het getal i . We hebben dan de graaf K_{k+1} , die zeker samenhangend is, en die evengraads is precies dan als k even is (elk punt heeft graad $(k+1)-1$).

(c) Laat $k \geq 3$. Als er géén kring bestaat die alle k -stenen bevat, hoeveel stenen (uitgedrukt in k) blijven er dan minimaal over? Beredeneer je antwoord. **Aanwijzing:** denk aan de graaf met alle gebruikte stenen.

Er blijven minimaal $(k+1)/2$ stenen over, en bij een geschikte keuze van dit aantal weggelaten stenen kan je ook effectief een kring leggen.

Redenering. Volgens deel (b) hebben we te maken met een oneven getal k . Als je een kring gelegd hebt, dan vormen de gelegde stenen samen een Euler graaf, en dus een evengraadse graaf. Elk punt heeft oneven graad k in de gegeven graaf (zonder lussen), en dus is elk punt zeker één lijn kwijt. Daarom zijn er tenminste $(k+1)/2$ stenen met ongelijke getallen overgebleven. Die schatting is scherp: laat je de stenen (lijnen) $(0,1)$, $(2,3)$, $\dots$, $(k-1,k)$ weg, dan is de restgraaf evengraads en duidelijk samenhangend. Met de overige kun je een kring leggen, en de dubbelstenen kun je daar nog steeds in kwijt als $k \geq 3$.

2. Het Prim Algoritme

(a) Algemene kennis.

(i) Wat is een *opspannende boom* in een graaf?

Een opspannende boom is een deelgraaf met dezelfde punten als de gegeven graaf ("opspannend"), en is bovendien een boom.
Onvolledig (dus fout) antwoord: het is een boom die de gegeven graaf opspant (zo'n boom kan "vreemde lijnen" gebruiken).

(ii) Welke eenvoudige voorwaarde(n) zijn nodig en voldoende voor een graaf om een opspannende boom te bezitten?

Een graaf heeft een opspannende boom precies dan als hij samenhangend is.

(b) Beschrijf het *algoritme van Prim* voor het vinden van een minimale opspannende boom in een gewogen graaf.

Oplossing:

- * beginboom T : kies 1 punt van de graaf.

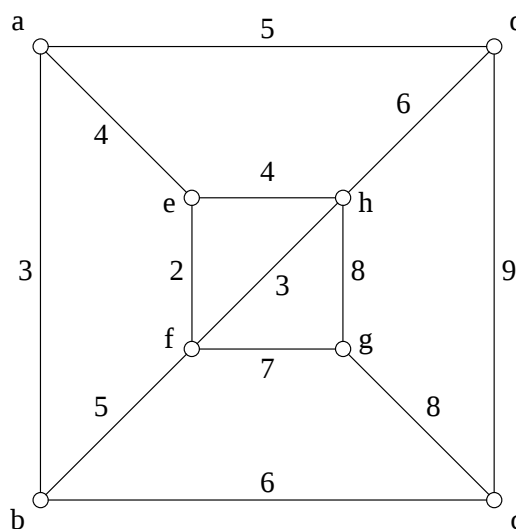
- * Herhaal zolang de boom T niet opspannend is:

- * zoek een lijn van minimaal gewicht tussen een punt van de boom en een punt daarbuiten

- * voeg die lijn en het andere eindpunt toe aan T .

Het resultaat is een opspannende boom als de gegeven graaf samenhangend is.

(c) Bepaal met dit algoritme een minimale opspannende boom in bijgevoegde graaf en geef het gewicht van de gevonden boom. Wanneer meerdere keuzes mogelijk zijn, kies dan voor de mogelijkheid die alfabetisch eerst komt.



Figuur bij Opgave 2: het Prim Algoritme

We beginnen met de triviale boom $\{a\}$ en voegen achtereenvolgens deze lijnen toe:

ab; ae; ef; fh; ad; bc; fg.

Gewicht: $ab+ae+ef+fh+ad+bc+fg=3+4+2+3+5+6+7=30$

3. Tellen

- (a) k verschillende pakjes moeten in een kring rond de kerstboom worden gelegd.

- (1) Op hoeveel manieren kun je de pakjes leggen?

Elke *rij* pakjes kan op unieke wijze worden weergegeven door middel van een bijectieve functie $Pakjes \rightarrow Posities$ (verschillende pakjes krijgen een verschillende positie). Daarbij is $\#Pakjes = k$. Het gaat hier dus om het (model)probleem van k -rangschikkingen van k , met $k!$ oplossingen.

Omdat de pakjes verschillend zijn, kun je een *gesloten kring* van k pakjes op k manieren openmaken tot herkenbaar verschillende rijen (kring in wijzerzin bekijken: welk pakje ligt eerst?). Het aantal oplossingen is dus k keer minder dan daarnet: $k! / k = (k-1)!$.

- (2) Bij nader order mogen pakjes A en B niet naast elkaar gelegd worden. Bepaal het aantal mogelijkheden wanneer $k=9$.

Maak voor de gelegenheid één superpak C in plaats van de aparte A en B. We hebben dan 8 pakjes in een kring te leggen. Dat kan volgens (1) op $8!/8=7!$ manieren. Vervang nu het superpak door (in wijzerzin) eerst A, dan B, te leggen of andersom. Dat geeft $2 \cdot 7!$ manieren om 9 pakjes in een kring te leggen met pakjes A en B wél naast elkaar. Negen pakjes kunnen we volgens (1) op $8!$ manieren in een kring leggen. Het aantal toegelaten oplossingen is dus:

$$\# \text{ mogelijk} - \# \text{ verboden} = 8! - 2 \cdot 7! = 40320 - 10080 = 30240.$$

Een formule met toelichting volstaat in beide onderdelen a1) en a2).

- (b) Laat zien dat de volgende formule juist is door op twee manieren een zelfde bedrag te tellen. Geef duidelijk aan wat je telt, en hoe je telt:

$$\binom{m+n}{p} = \sum_{i=0}^p \binom{m}{i} \binom{n}{p-i}.$$

Links tel je het aantal manieren om p verschillende dingen te kiezen uit een geheel met $m+n$ dingen. Rechts denk je m dingen in een verzameling A en n andere dingen in een verzameling B. Dan kun je het volgende alternatieve keuzeproces doorlopen: bepaal eerst welk aantal i (met $0 \leq i \leq p$) van dingen je uit A wil halen; de overige $p-i$ haal je dan uit B. Voor gegeven i is het aantal keuzes

$$\binom{m}{i} \cdot \binom{n}{p-i}.$$

Deze aantallen moeten worden opgeteld voor $i=0, \dots, p$.

- (c) Zeventien identieke knikkers worden verdeeld over zes verschillende potjes, waarbij precies twee potjes leeg dienen te blijven en precies één potje een oneven aantal knikkers moet bevatten. Hoeveel mogelijke resultaten zijn er? Geef toelichting bij de uitkomst. **Aanwijzing:** tel aan de hand van een keuzeproces dat in enkele fasen verloopt.

Een geschikt keuzeproces gaat als volgt: kies eerst twee potjes die leeg blijven. Dat kan op

$$\binom{6}{2} = 15$$

manieren. Uit de resterende vier potjes kies je één waar een oneven aantal knikkers in moet. Dat kan op

$$\binom{4}{1}=4$$

manieren. Leg één knikker in het "oneven" potje en twee knikkers in de drie resterende potjes. Er resten $17-7=10$ knikkers, die nu willekeurig met twee tegelijk over vier potjes verdeeld mogen worden. We verdelen in feite vijf "dubbelknikkers" over vier potjes. Dat kan op

$$\binom{5+4-1}{5}=56$$

manieren. Het totale aantal mogelijkheden is nu het product

$$\binom{6}{2} \cdot \binom{4}{1} \cdot \binom{5+4-1}{5} = 15 \cdot 4 \cdot 56 = 3360.$$

4. Recurrente betrekkingen

(a) Los de volgende *recurrente betrekking* op:

$$a_n - 9a_{n-2} = 8 \cdot 3^n \quad (a_0 = 9/4, a_1 = 3/4).$$

(Dictaat Open Universiteit opgave 3.34(i) p. 94)

De k.v. is $(r+3)(r-3)=0$ met wortels ± 3 .

Homogene oplossing: $a_n^{(h)} = k_1 \cdot (-3)^n + k_2 \cdot 3^n$.

Particuliere oplossing: Normale voorstel is $a_n^{(p)} := l \cdot 3^n$, maar dat werkt hier niet (3 is wortel van de k.v.). Correcte voorstel: $a_n^{(p)} := l \cdot n \cdot 3^n$. Invullen in de recurrente betrekking geeft $l=4$.

Algemene oplossing: $a_n^{(h)} + a_n^{(p)}$.

Beginvoorwaarden:

$$a_0 = k_1 \cdot (-3)^0 + k_2 \cdot 3^0 \text{ geeft (1) } k_1 + k_2 = 9/4.$$

$$a_1 = k_1 \cdot (-3)^1 + k_2 \cdot 3^1 + 4 \cdot 3^1 \text{ geeft (2) } -k_1 + k_2 = -15/4.$$

Uit (1)+(2) halen we $k_2 = -3/4$. Dit in (1) geeft $k_1 = 3$.

Oplossing:

$$a_n = 3 \cdot (-3)^n - 3/4 \cdot 3^n + 4 \cdot n \cdot 3^n = ((-1)^n \cdot 3 - 3/4 + 4n) \cdot 3^n.$$

(b) Los de volgende recurrente betrekking op met behulp van *genererende functies*.

$$a_n - 3a_{n-2} = 0 \quad (n \geq 2), \quad a_0 = 1, a_1 = 1.$$

Bepaal met de gevonden formule de getallen a_9 en a_{10} .

Laat $A := \sum_{n=0}^{\infty} a_n X^n$ de genererende functie zijn voor de oplossing $a_n)_{n=0}^{\infty}$.

Invullen in de recurrente betrekking ($n \geq 2$) geeft

$$\sum_{n=2}^{\infty} a_n X^n - 3 \sum_{n=2}^{\infty} a_{n-2} X^n = 0.$$

Hieruit volgt dat $A - a_0 - a_1 X - 3X^2 A = 0$ en dus

$$A = (a_0 + a_1 X) / (1 - 3X^2) = (1 + X) \cdot (1 - 3X^2)^{-1} = \sum_{n=0}^{\infty} 3^n X^{2n} + \sum_{n=0}^{\infty} 3^n X^{2n+1}.$$

Dan is $a_k = 3^{k/2}$ als k even is en $a_k = 3^{(k-1)/2}$ als k oneven is.

Dat geeft $a_9 = 3^4 = 81$ en $a_{10} = 3^5 = 243$.

(c) Een stratenmaker heeft twee soorten stenen tot zijn beschikking: dubbele en enkele lengte. Hij moet rijen leggen van lengte n , waarbij hij naar eigen smaak dubbele en enkele steenlengtes door elkaar mag gebruiken. Zij a_n het aantal mogelijkheden om een lengte n te leggen.

(i) Geef de waarde van a_1 , a_2 , en a_3 .

$a_1 = 1$ (leg een steen van enkele lengte).
 $a_2 = 2$ (twee enkele stenen of een dubbele).
 $a_3 = 3$ (drie enkele of enkele+dubbele of dubbele+enkele).

- (ii) Stel met een zorgvuldige redenering een recurrente betrekking op voor a_n ($n \geq 3$).

Zij $n \geq 3$. Een rij stenen van lengte n kan eindigen met een enkele of een dubbele steen. In het eerste geval zijn er a_{n-1} rijen mogelijk, in het tweede geval zijn er a_{n-2} rijen mogelijk. De mogelijkheden overlappen niet. Dus: $a_n = a_{n-1} + a_{n-2}$.

Je hoeft de aantallen niet verder te berekenen.

5. Coderingstheorie

- (a) Algemene kennis:

- (i) Wat bedoelt men met de *Hamming afstand* tussen twee woorden?

De Hamming afstand tussen twee woorden is het aantal posities waarin deze woorden verschillen.

- (ii) Wat bedoelt men met *maximum likelihood decoding*?

Met maximum likelihood decoding wordt een ontvangen woord omgezet naar het unieke, meest nabije codewoord.

Als er meerdere codewoorden tegelijk meest nabij zijn, dan is er geen decoding.

- (b) Gegeven is een *generator matrix* G voor een lineaire, binaire code.

- (i) Ga na hoeveel *informatie-* en *check symbolen* er zijn, en

- (ii) *codeer* het meegegeven bericht $\vec{b}$.

$$G := \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}, \quad \vec{b} := (1, 0, 1).$$

De matrix is 3 bij 7, dus $k=3$ informatiebits, $n=7$ woordlengte, en $m=4$ checksymbolen.

Coderen: $\vec{b} \cdot G = 1\text{e rij} + 3\text{e rij van } G = (1, 0, 1, 1, 1, 0, 0)$.

Merk op dat coderen met deze matrix de informatiebits vooraan plaatst.

- (c) (Vervolg deel (b)) Schrijf de bijhorende *check matrix* H uit.

G heeft een 3×3 eenheidsmatrix vooraan, gevolgd door een 3×4 blok. We vinden H door een 4×4 eenheidsmatrix achteraan te plaatsen, met daarvoor het gespiegelde 4×3 blok. Normaal moet dit blok overal een extra "-" krijgen, maar bij binair rekenen heeft dat geen effect.

$$H := \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

- (d) (Vervolg delen (b) en (c)) Een gecodeerd bericht $\vec{w} := (1, 0, 0, 0, 1, 0, 1)$ wordt ontvangen. *Decodeer* dit bericht na een eventuele fout te hebben verbeterd, of geef aan waarom de fout niet kan worden verbeterd.

Syndroom berekenen: $H \cdot \vec{w} = (1, 1, 1, 0) = 2\text{e kolom van } H$. Er is dus een fout in w , en wel op de 2e positie. Correctie: $(1, 1, 0, 0, 1, 0, 1)$. Decoding: eerste drie bits, dus $(1, 1, 0)$.

Normering

1a:	4	2a:	4	3a:	5	4a:	5	5a:	4	6:	20
b:	5	b:	3	b:	5	b:	5	b:	4		
c:	5	c:	5	c:	5	c:	4	c:	3		
								d:	4		
	14		12		15		14		15		20

(Multiplechoice: -3 ptn per fout)

Totaal te behalen = 90

Cijfer = Totaal/10 + 1

Naam:

Studierichting en jaar:

25-05-09

6. Multiple-choice gedeelte

- Dit blad moet in het tentamen formulier worden ingeleverd.
- In dit multiple-choice gedeelte is er per vraag precies één alternatief juist én (meest) volledig.

6.1. Het *Handelsreizigersprobleem* is het probleem om in een gewogen graaf

- o een pad van minimaal gewicht te vinden tussen twee punten.
- o een rondwandeling langs alle punten te vinden van minimaal gewicht. ➡ DEZE
- o een volledige toewijzing te vinden van minimaal gewicht.
- o een Hamilton-cykel te vinden van minimaal gewicht.

Commentaar:

Opties 1 en 3 zijn gerelateerd aan, respectievelijk, het Dijkstra algoritme en de Hongaarse methode. Optie 4 is erg beperkt: veel samenhangende gewogen grafen hebben geen Hamilton-cykels, maar er is wel een zinvolle oplossing voor het Handelsreizigersprobleem.

6.2. De stelling van *Berge-Petersen* zegt:

- o Een koppeling F is volledig precies dan als F alle punten verzadigt.
- o Een koppeling F is volledig precies dan als er geen vergrotend F –wisselpad bestaat.
- o Een koppeling F is maximum precies dan als er geen vergrotend F –wisselpad bestaat. ➡ DEZE
- o Elke twee koppelingen F, F' met $\#F < \#F'$ kunnen in balans worden gebracht door uitwisselen van lijnen.

Commentaar:

Optie 1 is de definitie van volledig, en optie 4 is een gevolg van de stelling van Berge-Petersen. Optie 2 is i.h.a. een foute bewering.

6.3. De *Hongaarse methode* betreft de constructie van

- o een maximum toewijzing in een tweedelingsgraaf. ➡ DEZE
- o een latijns vierkant uit een latijnse rechthoek.
- o een maximale wisselboom in een tweedelingsgraaf.
- o een maximum stroom in een netwerk.

Commentaar:

Latijnse vierkanten (optie 2) hebben te maken met een hele reeks van toewijzingen. Wisselbomen (optie 3) zijn enkel een onderdeel van de Hongaarse methode, niet het doel. Stromen in een netwerk (optie 4) hebben geen direct verband met het onderwerp.

6.4. We willen weten hoeveel *open kettingen* we kunnen maken met n kralen als er k kleuren beschikbaar zijn. De kralen zijn eender op de kleur na. Hoe beschrijf je dit probleem:

- o Een n –kombinatie van k elementen?
- o Een n –herhalingskombinatie van k elementen?
- o Een n –rangschikking van k elementen?
- o Een n –herhalingsrangschikking van k elementen? ➡ DEZE
- o Geen van deze beschrijvingen voldoet.

Commentaar:

Het gaat hier om een volgorde van kleuren, waarbij kleuren mogen herhaald worden. Daarmee is optie 4 de enige die juist is.

6.5. Het aantal oplossingen van $x_1 + 2 \cdot x_2 + \dots + n \cdot x_n = k$, met $x_1, \dots, x_n$ geheel en niet-negatief, stemt overeen met

- o het aantal verdelingen van k eendere knikkers in n eendere porties.
- o het aantal verdelingen van k eendere knikkers in eendere porties van grootte $\leq n$. ➡ DEZE
- o het aantal verdelingen van n eendere knikkers in k eendere porties.
- o het aantal verdelingen van n eendere knikkers in eendere porties van grootte $\leq k$.

Commentaar:

Bij opties 3 en 4 is de rol van n en k verwisseld. De term $i \cdot x_i$ komt overeen met het aantal knikkers in x_i hoopjes van grootte i : optie 2 dus.

N.B.: met "eendere porties" wordt bedoeld dat de porties, afgezien van hun grootte, niet te onderscheiden zijn.

6.6. Het principe van *inclusie-exclusie* in een "universum" U stelt dat het aantal elementen in U dat aan één van t condities voldoet, gelijk is aan:

- o de alternerende som $\sum_{1 \leq i \leq t} (-1)^i S_i$, waar S_i de som is over alle i -tallen condities van de aantallen elementen die deze i condities overtreden.
- o de alternerende som $\sum_{0 \leq i \leq t} (-1)^i S_i$, waar S_i de som is over alle i -tallen condities van de aantallen elementen die aan deze i condities voldoen. ➡ DEZE
- o de alternerende som $\sum_{1 \leq i \leq t} (-1)^i S_i$, waar S_i het aantal elementen is dat precies i condities overtreedt.
- o de alternerende som $\sum_{0 \leq i \leq t} (-1)^i S_i$, waar S_i het aantal elementen is dat aan precies i condities voldoet.

Commentaar:

Je telt per i -tal condities hoeveel elementen aan deze condities voldoen (exit opties 1 en 3), en die bedragen moet je botweg optellen voor alle i -tallen. In de volgende fase (met $(i+1)$ -tallen) wordt gecorrigeerd voor dubbelstellingen. Optie 4 is dus ook fout.

6.7. Een *perfect 3 fouten verbeterende code* is een code waarbij

- o voor ieder woord met ten hoogste drie fouten een uniek codewoord bestaat op afstand ten hoogste drie.
- o voor ieder woord een uniek codewoord bestaat op afstand ten hoogste drie. ➡ DEZE
- o voor ieder woord een gegarandeerd correcte decoding bestaat, zodat ten hoogste drie fouten hoeven te worden gecorrigeerd.
- o ieder woord met ten hoogste drie fouten correct kan worden gecorrigeerd.

Commentaar:

Opties 1 en 4 geven allebei weer wat men bedoelt met "een 3 fouten corrigerende code", waarbij het "perfecte" aspect buiten beschouwing valt. Optie 3 is onzin: een decoding is enkel de meest waarschijnlijke en is niet gegarandeerd correct; op het aantal fouten heb je bovendien geen vat.