Universiteit Utrecht
Betafaculteit

Examen Discrete Wiskunde I en II op donderdag 4 juli 2019, 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.
- Het examen omvat 9 opgaven met in totaal 16 (deel)opgaven.
- Bij de opgaven hieronder mag u er van uitgaan dat u een polynomiaal algoritme voor Max Flow, Min cost max flow, Kortste pad, Minimale Opspannende boom beschikbaar hebt. Wanneer u dit algoritme wilt toepassen hoeft u alleen aan te geven op welke instantie dit moet gebeuren.
- Op de vragen 3a, 7b, 7c en 7d kunnen maximaal 2 punten worden gescoord; op de vragen 1, 7a en 7d kunnen maximaal 3 punten worden gescoord en op alle overige vragen kunnen maximaal 4 punten worden gescoord. Totaal 53 punten.
- Wanneer gevraagd wordt om een probleem op een bepaalde manier op te lossen, dan wordt iedere andere manier volledig fout gerekend. Enumeratie levert eveneens niets op.
- Een rekenmachine is niet nodig en derhalve verboden. Eventuele rekenfouten worden niet bestraft, tenzij het de spuigaten uitloopt (dit ter beoordeling van de nakijker).

Succes!

==

Opgave 1.
Gegeven is een standaard pak kaarten (van ieder van de vier kleuren Schoppen, Harten, Ruiten en Klaveren zijn er dertien kaarten). Hieruit worden willekeurig 13 kaarten getrokken; in vaktermen wordt dit een *hand* genoemd. Er wordt gezegd dat een hand een *renonce* bevat indien van één kleur geen enkele kaart aanwezig is. Bereken de kans hierop. U mag er bij de berekening vanuit gaan dat de kans dat de hand van twee (of zelfs drie) kleuren geen enkele kaart bevat 0 is.

Opgave 2
Drie personen (genaamd A, B, C) zijn omgekocht met 1000 identieke goudstukken; deze willen ze gaan verdelen. Omdat A belangrijker is dan B, en B belangrijker is dan C, moet A meer goudstukken ontvangen dan B en moet B meer goudstukken ontvangen dan C. **Geef een formule (zonder sommatie)** voor het aantal mogelijkheden om die goudstukken te verdelen, waarbij C eventueel niets krijgt.

Opgave 3.
Los de onderstaande recurrente betrekkingen op met de JBF methode (karakteristieke vergelijking, particuliere oplossing, enz.). U mag eerder gevonden resultaten uiteraard hergebruiken. Bij (b) zijn de beginwaarden niet gegeven; hier hoeft u de constanten niet uit te rekenen.

$(a)\quad a_n = 4a_{n-1} - 4a_{n-2}\quad$ voor $n \geq 1$ met $a_0 = 1$ en $a_1 = 4$.

$(b)\quad a_n = 4a_{n-1} - 4a_{n-2} + 2 \cdot 2^n + n\quad$ voor $n \geq 2$ met a_0 en a_1.

Opgave 4.
Bepaal het aantal gehele getallen n, met $1 \leq n \leq 1000$ dat tegelijkertijd aan de volgende eigenschappen voldoet:

1. n moet deelbaar zijn door 3.
2. n moet deelbaar zijn door 4, maar niet door 16.
3. n mag niet deelbaar zijn door 5.

Volledige enumeratie kost alleen tijd en levert geen punten op.

Opgave 5
Op een verjaardagstaart staan n kaarsjes die de jarige moet uitblazen. De jarige heeft goed geoefend, en bij iedere keer blazen gaan er minstens twee kaarsjes uit; verder geldt dat als er k (met $k \geq 2$) kaarsjes branden, dan is de kans dat er precies j kaarsjes worden uitgeblazen gelijk aan $1/(k-1)$, voor $j = 2, \ldots, k$; wanneer er nog één kaarsje brandt, dan wordt dat zeker uitgeblazen. Bepaal het verwachte aantal malen dat de jarige moet blazen om alle kaarsjes uit te blazen.

De verwachte waarde van een stochast X die met kans $P(X = x)$ een waarde x kan aannemen is gelijk aan $\sum_{x \in S} x \cdot P(X = x)$, waarbij S de verzameling van waarden is die X kan aannemen; voor een gewone dobbelsteen vind je dan dat de verwachte waarde gelijk is aan $1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + \ldots + 6 \cdot \frac{1}{6} = 3,5$.

Opgave 6
Gegeven is een verzameling van 10 niet-negatieve, gehele getallen $a_1, \ldots, a_{10}$, waarbij geldt $a_i \leq 100$ $(i = 1, \ldots, 10)$. Toon aan dat het altijd mogelijk is om twee verschillende deelverzamelingen S en T van $\{1, \ldots, 10\}$ te vinden zodanig dat

$$\sum_{j \in S} a_j = \sum_{j \in T} a_j$$

Is het ook altijd mogelijk om twee disjuncte deelverzamelingen S en T te vinden?

Opgave 7.
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.

Stel dat je een formule voor de Pattern Inventory kent. Definieer $Inve(w(rood), w(blauw), w(wit))$ als de waarde van de pattern inventory wanneer je getal of symbool $w(rood)$, enz. invoert (bijv. $Inve(2, 2, 2)$ geeft de waarde aan wanneer iedere kleur waarde 2 krijgt). Bij de onderstaande vragen moet je aangeven hoe je het gewenste antwoord kunt bepalen door een goede waarde of symbool in te vullen voor $w(rood)$, $w(blauw)$ en $w(wit)$. Je hoeft het antwoord zelf niet uit te rekenen (mag wel). Je mag nooit meer symbolen gebruiken dan aangegeven. Enumeratie levert niets op en kost alleen tijd. **Motiveer steeds uw antwoord**.

(b) Bepaal het aantal equivalentieklassen dat geen rode kralen bevat.

(c) Bepaal het aantal equivalentieklassen met precies één witte kraal die meer blauwe dan rode kralen bevatten. Je mag hierbij in totaal maximaal 1 symbool gebruiken.

(d) Bepaal het aantal equivalentieklassen met minstens één rode en minstens één witte kraal.

(e) Bepaal het aantal equivalentieklassen met een even aantal blauwe, een even aantal rode, en een even aantal witte kralen.

Opgave 8.
Gegeven zijn n munten, die allemaal een verschillende, positieve, geheeltallige waarde w_j ($j = 1, \ldots, n$) hebben.

(a) Geef een algoritme dat op efficiënte wijze (dus beter dan volledige enumeratie) nagaat of het mogelijk is om een gegeven bedrag Q precies te betalen.

(b) Geef een algoritme dat op efficiënte wijze (dus beter dan volledige enumeratie) nagaat op hoeveel verschillende manieren je het bedrag Q precies kunt betalen.

Opgave 9.
Vroeger was de thuiszorg simpel: de hulp kwam langs wanneer hij/zij tijd had (de patiënt was toch altijd thuis), hielp de patiënt, maakte een praatje, dronk een kop koffie en vertrok weer. Tegenwoordig is dit proces sterk gestroomlijnd: voor iedere patiënt is afgesproken wanneer hij/zij bezocht wordt, en wanneer de tijd verstreken is, dan moet de hulp er van door. Om dit proces robuust te plannen wordt uw hulp gevraagd. Van iedere klant j ($j = 1, \ldots, n$) is bekend gedurende welk interval deze geholpen moet worden (noteer dit als $[a_j, b_j]$, waarbij a_j de begintijd en b_j de eindtijd is). Verder is uiteraard bekend waar patiënt j zich bevindt (noteer dit als plaats v_j) en wat de reistijden zijn tussen ieder tweetal plaatsen v_i en v_j (noteer deze als d_{ij}).

Om deze patiënten te helpen is een groep van k hulpen beschikbaar. Deze zijn allemaal even bekwaam (het maakt dus niet uit welke hulp welke patiënt verzorgt). Ze beginnen allemaal op tijdstip 8.00 op de gemeenschappelijke verzamelplaats en komen na afloop van hun werkdag daar ook weer terug (om de administratie in orde te brengen). Bij de verzorging van patiënt j ($j = 1, \ldots, n$) is precies één hulp nodig gedurende de volledige periode $[a_j, b_j]$.

(a) Formuleer een algoritme dat in **polynomiale** tijd een toegelaten oplossing vindt waarin zoveel mogelijk patiënten worden geholpen.

(b) Gelukkig is men bezig om nieuwe hulpen op te leiden. Persoon X is zo'n stagair, en hij/zij kijkt een dagje mee bij interessante patiënten. Bepaal voor X een toegelaten schema met maximale opbrengst. De opbrengst van meekijken bij het helpen van patiënt j bedraagt c_j; X moet daar dan de gehele periode $[a_j, b_j]$ bij aanwezig zijn. U mag aannemen dat bij (a) een schema gevonden is waarin iedere patiënt wordt bezocht.