Huiswerk week 6

Opgave 1.
In deze opgave bewijzen we de *stelling van Cayley*: De volledige graaf K_n heeft n^{n-2} opspannende bomen.

(i) Zij $T_{n,k}$ het aantal bossen in K_n met precies k samenhangscomponenten, waarbij de knopen $\{1, \dots, k\}$ in verschillende componenten liggen.

Laat zien dat $T_{n,k}$ aan de recursie

$$T_{n,k} = \sum_{i=0}^{n-k} \binom{n-k}{i} T_{n-1,k-1+i}$$

voldoet, waarbij $T_{0,0} := 1$ en $T_{n,0} := 0$ voor $n > 0$.

(Hint: Ga na wat er gebeurt als de knoop 1 verwijderd wordt.)

(ii) Laat zien dat $T_{n,k} = kn^{n-k-1}$.

(iii) Concludeer dat het aantal opspannende bomen $T_{n,1} = n^{n-2}$ is.

Opgave 2.
Zij $G = (V, E)$ een samenhangende kant-gelabelde graaf.

(i) Stel dat de gewichten van de kanten van G alle verschillend zijn.

Bewijs dat G dan een unieke minimale opspannende boom heeft.

(ii) Zij T een minimale opspannende boom van G. Laat zien dat in het algoritme van Kruskal de kanten zo gekozen kunnen worden dat het algoritme T terug geeft (d.w.z. iedere minimale opspannende boom is een mogelijke output van het algoritme van Kruskal).

(Hint: Het is handig om de gewichten om kleine hoeveelheden ε te wijzigen en een soort continuiteitsargument toe te passen.)

Opgave 3. (Cameron: Chapter 11, opgave 4)
Een variatie op het algoritme van Kruskal om een minimale opspannende boom te construëren is het *algoritme van Prim*. Dit werkt als volgt:

- Begin met een kant e_0 van minimaal gewicht en laat $S := \{e_0\}$;
- laat een boom als volgt groeien:
  - kies een kant $e \in E \setminus S$ van minimaal gewicht met de volgende twee eigenschappen:
    1) e heeft een knoop gemeenschappelijk met een van de kanten in S;
    2) e verbindt knopen uit verschillende samenhangscomponenten in (V, S).

– voeg e aan S toe, d.w.z. vervang (V, S) door $(V, S \cup \{e\})$;

– stop het groeiproces als er $|V| - 1$ kanten in S zitten.

Laat zien dat het algoritme van Prim een minimale opspannende boom oplevert.

Opgave 4.
Implementeer een algoritme dat voor een kant-gelabelde graaf met n knopen, (gegeven door een $n \times n$-matrix van gewichten) een minimale opspannende boom berekent.
Zij $A_n \in \mathbb{Z}^{n \times n}$ de matrix gedefinieerd door

$$(A_n)_{i,j} := (i^2 - j^2)^2 \mod 199.$$

Pas je algoritme op de volledige graaf K_n met gewichtsmatrix A_n toe voor $n = 16, 32, 64$ en geef voor ieder geval de som der gewichten in een minimale opspannende boom aan.

Opgave 5.
De tabel hieronder geeft de afstanden tussen 25 steden in Nederland aan.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	47	47	46	139	123	86	111	114	81	164	67	126	73	18	147	190	176	63	141	78	20	109	65	70
2	47	0	89	92	162	134	100	125	156	57	184	20	79	87	30	132	207	175	109	168	77	40	151	107	103
3	47	89	0	25	108	167	130	103	71	128	133	109	154	88	65	129	176	222	42	127	125	67	66	22	41
4	46	92	25	0	132	145	108	78	85	116	157	112	171	63	64	154	151	200	17	102	113	59	64	31	66
5	139	162	108	132	0	262	225	210	110	214	25	182	149	195	156	68	283	315	149	234	217	159	143	108	69
6	123	134	167	145	262	0	37	94	230	83	187	124	197	82	119	265	183	59	128	144	57	103	209	176	193
7	86	100	130	108	225	37	0	57	193	75	250	111	179	45	82	228	147	96	91	107	49	66	172	139	156
8	111	125	103	78	210	94	57	0	163	127	235	141	204	38	107	232	125	153	61	50	101	91	142	109	144
9	114	156	71	85	110	230	193	163	0	195	135	176	215	148	132	155	237	285	102	187	192	134	40	54	71
10	81	57	128	116	214	83	75	127	195	0	236	41	214	104	72	182	217	124	133	177	26	61	180	146	151
11	164	184	133	157	25	187	250	235	135	236	0	199	147	220	178	58	309	340	174	259	242	184	168	133	94
12	67	20	109	112	182	124	111	141	176	41	199	0	73	103	49	141	226	165	130	184	67	56	171	127	123
13	126	79	154	171	149	197	179	204	215	214	147	73	0	166	109	89	289	238	188	247	140	119	220	176	144
14	73	87	88	63	195	82	45	38	148	104	220	103	166	0	69	215	123	141	46	81	79	53	127	94	129
15	18	30	65	64	156	119	82	107	132	72	178	49	109	69	0	146	192	172	81	150	74	16	127	83	88
16	147	132	129	154	68	265	228	232	155	182	58	141	89	215	146	0	306	306	171	256	208	162	183	139	91
17	190	207	176	151	283	183	147	125	237	217	309	226	289	123	192	306	0	243	135	50	191	176	213	183	218
18	176	175	222	200	315	59	96	153	285	124	340	165	238	141	172	306	243	0	187	203	98	156	264	231	246
19	63	109	42	17	149	128	91	61	102	133	174	130	188	46	81	171	135	187	0	85	111	76	81	48	83
20	141	168	127	102	234	144	107	50	187	177	259	184	247	81	150	256	50	203	85	0	151	134	166	133	168
21	78	77	125	113	217	57	49	101	192	26	242	67	140	79	74	208	191	98	111	151	0	58	177	143	148
22	20	40	67	59	159	103	66	91	134	61	184	56	119	53	16	162	176	156	76	134	58	0	123	85	90
23	109	151	66	64	143	209	172	142	40	180	168	171	220	127	127	183	213	264	81	166	177	123	0	44	92
24	65	107	22	31	108	176	139	109	54	146	133	127	176	94	83	139	183	231	48	133	143	85	44	0	48
25	70	103	41	66	69	193	156	144	71	151	94	123	144	129	88	91	218	246	83	168	148	90	92	48	0

Een matrix met deze afstanden in MAGMA format is te vinden in het bestand `nl_afstanden` op de webpagina voor deze cursus (`http://www.math.ru.nl/~souvi/dw1_09/dw1.html`).

(i) Vind een zo kort mogelijke Hamilton cykel door de 25 steden met behulp van het gretige algoritme (dat bij een stad begint en vervolgens naar de dichtst bij liggende stad loopt die nog niet bezocht is). Merk op dat het gretige algoritme voor verschillende beginsteden verschillende resultaten kan opleveren.

(ii) Vind een zo kort mogelijke Hamilton cykel door de 25 steden met behulp van het *Tweekeer-rond-de-boom algoritme.* Vergelijk het resultaat met de uitkomst van het gretige algoritme.

(iii) Vind een zo kort mogelijke Hamilton cykel door de 25 steden met behulp van een methode naar eigen keuze. Een lengte van meer dan $1250\,km$ is onvoldoende. Een lengte van minder dan $1220\,km$ levert je een bonus van 0.5 op je eindcijfer op.

Webpagina: http://www.math.ru.nl/~souvi/dw1_09/dw1.html