

# Programmieraufgaben 3

## Aufgabe 1

*Anmerkung: Für einen Graphen ex. keine Bipartition g.d.w. ein Kreis ungerader Länge existiert. Denn in einem Kreis ungerader Länge muss es eine Kante zwischen zwei Knoten der gleichen Partition geben, da ein gerader Pfad in einem Bipartiten Graphen in der gleichen Partition anfängt und endet.*

### Graph1

Executing BipartitionAlgorithm

@red\_nodes

label

1  
3  
4  
7  
8  
9  
11  
12  
14  
15  
16  
18  
20  
21  
22  
25  
27  
28  
31  
32  
33  
38  
41  
43  
44  
45  
47  
48  
49  
50  
52  
53  
54  
55  
59  
61  
62  
66

67  
70  
@blue\_nodes  
label

2  
5  
6  
10  
13  
17  
19  
23  
24  
26  
29  
30  
34  
35  
36  
37  
39  
40  
42  
46  
51  
56  
57  
58  
60  
63  
64  
65  
68  
69

@edges

		label	weight
70	46	0	5
52	46	1	9
52	30	2	4
66	30	3	6
66	69	4	7
70	69	5	1
44	69	6	7
44	63	7	4
66	63	8	4
1	63	9	7
1	65	10	4
67	65	11	9
67	69	12	1
67	42	13	10
70	42	14	9
61	42	15	8
61	19	16	1

67	19	17	6
50	19	18	10
50	58	19	8
62	58	20	7
62	64	21	8
67	64	22	9
38	64	23	3
38	58	24	4
38	19	25	6
38	63	26	8
38	34	27	2
70	34	28	5
59	34	29	5
59	17	30	5
70	17	31	3
48	17	32	8
48	63	33	2
48	56	34	8
66	56	35	6
44	56	36	4
53	56	37	8
53	68	38	7
20	68	39	9
20	42	40	1
20	40	41	5
67	40	42	1
59	40	43	9
53	40	44	3
52	40	45	3
45	40	46	6
45	68	47	9
45	46	48	7
45	35	49	9
70	35	50	2
67	35	51	5
50	35	52	6
48	35	53	4
44	35	54	6
41	35	55	9
41	68	56	9
41	42	57	7
41	65	58	5
41	56	59	10
41	40	60	3
41	17	61	7
41	69	62	6
41	36	63	3
70	36	64	6
50	36	65	2
55	36	66	6
55	64	67	2
55	60	68	9
66	60	69	2

15	60	70	4
15	58	71	3
15	42	72	9
15	24	73	1
66	24	74	3
48	24	75	9
31	24	76	6
31	64	77	9
31	60	78	8
31	57	79	1
62	57	80	10
1	57	81	10
9	57	82	4
9	69	83	4
9	60	84	6
9	51	85	2
66	51	86	4
61	51	87	1
70	51	88	10
11	51	89	2
11	42	90	8
11	30	91	9
11	37	92	5
67	37	93	1
52	37	94	7
49	37	95	9
49	60	96	5
49	6	97	7
54	6	98	7
54	58	99	3
54	56	100	6
54	57	101	9
54	39	102	8
50	39	103	9
3	39	104	8
3	46	105	7
3	34	106	2
3	30	107	6
3	29	108	10
66	29	109	6
61	29	110	1
59	29	111	5
54	29	112	8
49	29	113	8
48	29	114	9
43	29	115	9
43	68	116	3
43	63	117	5
43	58	118	7
43	57	119	5
43	46	120	7
43	17	121	5
43	60	122	3

43	23	123	6
62	23	124	4
49	23	125	1
38	23	126	7
31	23	127	2
4	23	128	4
4	58	129	1
4	36	130	2
4	29	131	5
4	26	132	4
50	26	133	2
49	26	134	3
48	26	135	7
28	26	136	4
28	65	137	6
28	63	138	4
28	29	139	6
28	19	140	6
28	37	141	8
28	68	142	4
28	40	143	8
28	58	144	2
28	24	145	9
28	17	146	1
28	5	147	10
66	5	148	1
55	5	149	10
47	5	150	10
47	68	151	4
47	58	152	6
47	30	153	10
47	69	154	1
47	37	155	6
47	6	156	6
47	63	157	2
47	19	158	7
47	64	159	8
47	39	160	4
47	36	161	10
47	35	162	2
47	29	163	3
47	26	164	6
47	24	165	2
47	17	166	4
47	13	167	9
48	13	168	6
45	13	169	9
22	13	170	10
22	63	171	3
22	51	172	1
22	56	173	5
22	17	174	8
22	46	175	2

22	39	176	8
22	64	177	8
22	6	178	9
22	57	179	10
22	58	180	1
22	36	181	10
22	42	182	3
22	10	183	6
70	10	184	7
52	10	185	10
50	10	186	8
21	10	187	8
21	65	188	2
21	63	189	8
21	58	190	1
21	36	191	6
21	23	192	5
21	19	193	3
20	10	194	2
12	10	195	10
12	58	196	6
12	56	197	2
12	46	198	10
12	39	199	3
12	17	200	10
12	40	201	7
12	26	202	6
12	51	203	8
12	36	204	7
12	63	205	5
12	2	206	7
62	2	207	9
52	2	208	2
50	2	209	9
32	2	210	8
32	64	211	4
32	51	212	9
32	42	213	10
32	69	214	8
32	29	215	8
32	24	216	5
32	19	217	7
32	5	218	4
14	2	219	6
14	65	220	6
14	60	221	2
14	56	222	2
14	46	223	5
14	42	224	1
14	40	225	7
14	26	226	7
14	69	227	1
14	39	228	9

14	35	229	7
14	51	230	8
14	68	231	2
14	6	232	6
14	29	233	7
14	24	234	4
14	57	235	10
14	13	236	5
67	2	237	7
18	2	238	9
18	40	239	7
18	46	240	3
18	35	241	1
18	26	242	8
18	23	243	9
18	51	244	1
18	17	245	9
20	2	246	5
27	2	247	1
27	69	248	9
27	64	249	9
27	58	250	7
27	51	251	8
27	35	252	10
27	29	253	7
27	46	254	6
27	39	255	10
27	26	256	4
27	23	257	6
27	6	258	7
43	2	259	4
25	2	260	10
25	60	261	4
25	39	262	8
25	35	263	7
25	58	264	8
25	56	265	7
25	5	266	1
25	42	267	1
25	46	268	1
25	17	269	5
25	23	270	9
25	13	271	6
25	6	272	9
47	2	273	6
66	2	274	1
12	42	275	9
12	6	276	10
55	10	277	10
11	10	278	6
33	10	279	3
33	65	280	2
33	68	281	2

33	64	282	1
33	42	283	9
33	6	284	8
33	56	285	2
33	30	286	7
33	17	287	1
59	10	288	4
7	10	289	6
7	19	290	5
7	17	291	2
7	46	292	4
7	69	293	4
7	39	294	4
7	42	295	10
7	60	296	9
7	26	297	9
7	51	298	4
7	34	299	8
7	68	300	9
45	10	301	2
44	10	302	7
1	10	303	1
22	19	304	10
20	13	305	2
62	13	306	6
54	13	307	6
38	13	308	4
53	13	309	3
9	13	310	8
3	13	311	6
38	5	312	1
15	5	313	5
54	5	314	1
48	5	315	3
41	5	316	5
20	5	317	10
38	26	318	10
20	26	319	10
11	26	320	7
8	26	321	2
8	69	322	2
8	58	323	5
8	46	324	3
8	34	325	1
8	30	326	7
8	68	327	3
8	64	328	10
8	60	329	3
8	65	330	3
4	51	331	9
4	30	332	6
4	63	333	4
4	24	334	7



4	68	335	5
4	69	336	3
4	19	337	8
4	65	338	4
4	39	339	5
44	23	340	10
55	23	341	4
66	23	342	8
67	23	343	4
54	23	344	1
52	23	345	2
3	23	346	4
1	23	347	4
43	39	348	9
43	37	349	2
43	36	350	4
38	29	351	1
45	29	352	6
62	29	353	4
41	29	354	2
52	29	355	9
50	29	356	9
44	29	357	3
53	29	358	6
9	29	359	6
3	19	360	8
3	17	361	8
3	51	362	7
3	58	363	6
62	39	364	2
44	39	365	4
67	39	366	8
66	39	367	5
31	39	368	9
11	39	369	8
9	39	370	3
1	39	371	7
54	36	372	7
54	34	373	3
54	19	374	4
54	17	375	2
52	6	376	4
45	6	377	2
31	6	378	7
9	6	379	4
66	6	380	3
67	6	381	3
11	6	382	1
48	6	383	4
1	6	384	10
49	64	385	1
49	68	386	7
49	34	387	8

49	51	388	7
49	42	389	8
49	24	390	7
49	19	391	3
1	37	392	1
61	37	393	1
70	37	394	7
31	37	395	2
16	37	396	3
16	68	397	8
16	63	398	10
16	40	399	6
16	35	400	4
16	34	401	4
16	24	402	10
16	60	403	2
16	30	404	7
16	64	405	1
16	65	406	3
16	56	407	3
15	37	408	9
11	58	409	2
11	56	410	10
11	34	411	7
11	68	412	1
11	36	413	1
11	17	414	1
48	51	415	10
45	51	416	2
44	51	417	7
31	51	418	7
20	51	419	10
15	51	420	7
9	30	421	8
9	46	422	7
9	58	423	9
9	34	424	2
9	36	425	8
9	19	426	5
9	64	427	9
44	57	428	5
52	57	429	5
48	57	430	3
31	40	431	5
31	46	432	8
31	19	433	10
44	24	434	8
70	24	435	1
45	24	436	5
61	24	437	10
15	17	438	6
15	46	439	2
15	68	440	8

61	60	441	2
1	60	442	4
20	60	443	6
55	65	444	2
55	63	445	2
66	36	446	6
48	36	447	4
38	35	448	8
1	35	449	10
66	35	450	8
45	58	451	5
45	34	452	10
1	40	453	5
20	17	454	1
20	69	455	1
20	63	456	10
20	64	457	10
20	19	458	1
67	68	459	8
50	68	460	6
38	68	461	10
1	68	462	7
53	58	463	6
53	64	464	1
53	34	465	8
53	30	466	5
67	56	467	9
62	56	468	10
59	56	469	10
50	56	470	5
38	56	471	5
1	56	472	8
48	58	473	2
48	42	474	4
44	17	475	8
59	58	476	1
59	69	477	3
52	34	478	6
66	34	479	9
38	30	480	10
38	69	481	3
38	65	482	3
1	64	483	2
50	64	484	5
44	64	485	8
62	42	486	2
62	30	487	8
67	58	488	7
66	58	489	6
1	58	490	5
50	46	491	9
66	19	492	9
52	19	493	4

44	19	494	4
44	42	495	4
1	42	496	8
67	30	497	6
66	65	498	10
70	65	499	8

is valid partitioning? 1

## Graph2

Executing BipartitionAlgorithm

@red\_nodes

label

1

4

5

6

11

12

13

14

16

24

25

26

28

31

33

34

35

36

37

39

42

46

47

50

@blue\_nodes

label

2

3

7

8

9

10

15

17

18

19

20

21

22

23  
27  
29  
30  
32  
38  
40  
41  
43  
44  
45  
48  
49

@edges

		label	weight
50	18	0	3
42	18	1	9
42	49	2	5
46	49	3	10
46	8	4	1
50	8	5	4
42	8	6	4
16	8	7	8
16	48	8	6
50	48	9	10
33	48	10	5
33	32	11	6
47	32	12	2
47	17	13	7
31	17	14	6
31	43	15	6
25	43	16	7
25	40	17	1
50	40	18	7
46	40	19	6
5	40	20	5
5	30	21	4
39	30	22	6
39	49	23	6
39	41	24	5
50	41	25	8
16	41	26	6
26	41	27	9
26	48	28	1
26	44	29	2
4	44	30	9
4	45	31	7
47	45	32	3
46	45	33	3
42	45	34	1
36	45	35	2
36	15	36	6
50	15	37	3

33	15	38	4
37	15	39	2
37	44	40	6
37	43	41	3
37	32	42	9
37	22	43	6
42	22	44	10
34	22	45	4
34	23	46	6
50	23	47	7
39	23	48	6
35	23	49	4
35	49	50	3
35	20	51	10
39	20	52	7
34	20	53	10
33	20	54	5
28	20	55	5
28	43	56	9
28	18	57	8
28	2	58	2
37	2	59	7
33	2	60	2
31	2	61	1
4	2	62	10
14	2	63	1
14	41	64	5
14	30	65	1
14	17	66	2
14	8	67	5
14	9	68	7
50	9	69	2
39	9	70	5
35	9	71	8
33	9	72	9
31	9	73	9
47	9	74	5
37	9	75	10
25	9	76	8
46	9	77	10
36	9	78	9
11	9	79	9
11	45	80	7
11	44	81	1
11	40	82	1
11	30	83	9
11	27	84	3
50	27	85	1
34	27	86	5
13	27	87	2
13	44	88	8
13	32	89	1
13	15	90	3

13	23	91	9
13	38	92	4
47	38	93	7
24	38	94	8
24	45	95	6
24	27	96	1
24	17	97	1
24	49	98	1
24	40	99	5
24	3	100	6
33	3	101	1
12	3	102	8
12	44	103	4
12	32	104	5
12	23	105	8
12	20	106	2
12	15	107	10
12	49	108	4
12	48	109	9
12	30	110	9
12	45	111	6
12	10	112	8
46	10	113	6
34	10	114	4
28	10	115	1
16	10	116	1
11	10	117	1
24	10	118	6
36	10	119	6
6	10	120	5
6	22	121	5
6	20	122	8
6	8	123	9
6	19	124	7
47	19	125	4
42	19	126	9
34	19	127	6
31	19	128	8
26	19	129	4
33	19	130	4
37	19	131	6
6	15	132	10
6	17	133	2
6	30	134	8
6	41	135	10
6	27	136	5
6	45	137	7
6	3	138	4
6	2	139	3
5	10	140	9
12	9	141	1
47	3	142	5
13	3	143	2

46	3	144	2
11	3	145	8
28	3	146	7
26	3	147	7
42	3	148	7
1	3	149	10
1	30	150	8
1	15	151	3
1	8	152	9
1	32	153	9
1	2	154	9
1	43	155	9
1	49	156	6
1	20	157	9
1	41	158	3
1	7	159	8
28	7	160	2
25	7	161	4
47	7	162	7
35	7	163	3
13	7	164	4
5	7	165	7
28	38	166	3
13	49	167	5
13	18	168	7
13	30	169	7
13	8	170	5
13	2	171	5
36	27	172	3
33	27	173	7
5	27	174	8
11	22	175	3
11	15	176	6
11	18	177	7
11	29	178	7
46	29	179	6
33	29	180	1
5	29	181	3
11	21	182	8
47	21	183	10
46	21	184	5
35	21	185	1
26	21	186	5
42	21	187	2
36	21	188	9
11	32	189	8
11	8	190	10
11	2	191	2
39	2	192	10
35	2	193	9
28	23	194	5
28	15	195	2
37	20	196	8



26	20	197	7
5	20	198	8
35	30	199	8
35	44	200	4
35	15	201	5
35	8	202	9
46	23	203	1
34	30	204	6
34	49	205	5
34	40	206	1
26	22	207	9
25	22	208	5
5	22	209	8
46	22	210	4
37	45	211	7
37	8	212	8
31	15	213	7
5	15	214	10
36	18	215	6
36	8	216	9
39	45	217	1
26	45	218	2
16	45	219	4
4	40	220	10
4	30	221	1
4	8	222	7
4	32	223	7
4	17	224	1
47	44	225	3
50	44	226	2
46	44	227	4
33	44	228	2
26	30	229	3
26	17	230	7
26	18	231	5
25	41	232	8
39	40	233	5
39	32	234	5
31	30	235	5
16	30	236	9
42	30	237	5
46	30	238	4
5	18	239	1
5	32	240	8
5	43	241	6
5	17	242	5
25	32	243	7
25	49	244	1
47	43	245	8
42	43	246	8
31	32	247	5
46	32	248	8
16	32	249	10

is valid partitioning? 1

Graph3

Executing BipartitionAlgorithm

@red\_nodes

label

96

97

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195

196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248

```

249
250
@blue_nodes
label
46
79
@edges
      label weight
100  46    0   10
97   46    1    6
97   79    2    1
96   79    3   10

```

```

is valid partitioning? 0
Es liegt ein Kreis ungerader Länge vor:
100 96 79 97 46 100

```

## Graph4

```

Executing BipartitionAlgorithm
@red_nodes
label
1
3
12
13
17
22
24
25
28
30
31
34
38
39
41
43
44
45
46
47
49
51
52
55
56
57
59
61
62
63
68

```

69  
70  
72  
76  
77  
79  
82  
85  
87  
88  
93  
94  
98  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139

140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
@blue\_nodes  
label  
2  
4  
5  
6  
7  
8  
9  
10  
11  
14  
15  
16  
18  
19  
20  
21  
23  
26  
27  
29  
32  
33  
35  
36  
37  
40  
42  
48  
50  
53  
54  
58  
60  
64  
65  
66  
67  
71  
73  
74

75  
78  
80  
81  
83  
84  
86  
89  
90  
91  
92  
95  
96  
97  
99

@edges

		label	weight
100	65	0	10
88	65	1	8
88	95	2	4
98	95	3	4
98	99	4	3
62	99	5	9
62	83	6	10
70	83	7	7
70	84	8	6
63	84	9	3
63	89	10	8
94	89	11	6
94	23	12	2
100	23	13	5
98	23	14	8
69	23	15	3
69	92	16	1
94	92	17	5
43	92	18	1
43	90	19	3
100	90	20	5
88	90	21	2
3	90	22	5
3	99	23	9
3	95	24	1
3	91	25	2
100	91	26	5
98	91	27	5
93	91	28	7
93	96	29	10
55	96	30	2
55	99	31	10
55	97	32	7
30	97	33	7
30	90	34	4
30	75	35	3



98	75	36	10
94	75	37	7
85	75	38	5
85	96	39	9
85	97	40	2
85	54	41	1
93	54	42	7
82	54	43	6
82	96	44	10
82	92	45	6
82	67	46	4
79	67	47	1
79	92	48	7
79	91	49	7
79	89	50	5
79	83	51	8
79	81	52	7
87	81	53	8
87	89	54	4
87	33	55	5
88	33	56	6
57	33	57	9
57	95	58	9
57	92	59	5
57	91	60	9
57	75	61	9
57	9	62	2
93	9	63	8
87	9	64	4
70	9	65	3
62	9	66	7
59	9	67	7
59	92	68	2
59	64	69	3
93	64	70	6
88	64	71	9
79	64	72	10
72	64	73	10
72	81	74	2
72	67	75	6
72	29	76	6
49	29	77	1
49	91	78	6
49	90	79	2
49	67	80	5
49	53	81	8
77	53	82	4
77	96	83	2
77	83	84	6
77	8	85	3
98	8	86	4
79	8	87	5
68	8	88	7

68	97	89	8
68	81	90	7
68	64	91	1
68	35	92	8
93	35	93	1
82	35	94	8
63	35	95	3
57	35	96	8
56	35	97	8
56	91	98	7
56	78	99	8
94	78	100	5
88	78	101	9
82	78	102	9
61	78	103	3
61	89	104	9
61	83	105	5
61	81	106	3
61	80	107	6
88	80	108	5
94	80	109	5
62	80	110	6
39	80	111	1
39	99	112	3
39	97	113	8
39	92	114	8
39	78	115	8
39	75	116	8
39	71	117	8
100	71	118	1
88	71	119	3
82	71	120	7
1	71	121	9
1	97	122	10
1	91	123	1
1	89	124	6
1	73	125	2
100	73	126	3
94	73	127	4
79	73	128	3
63	73	129	1
85	73	130	4
68	73	131	10
55	73	132	8
1	66	133	1
100	66	134	6
82	66	135	8
79	66	136	7
51	66	137	2
51	99	138	6
51	71	139	1
51	58	140	5
87	58	141	5

79	58	142	10
76	58	143	5
76	95	144	5
76	86	145	4
100	86	146	5
46	86	147	5
46	84	148	2
46	83	149	1
46	78	150	1
46	74	151	6
98	74	152	9
88	74	153	4
82	74	154	3
77	74	155	2
55	74	156	6
25	74	157	1
25	42	158	5
79	42	159	1
68	42	160	1
61	42	161	10
56	42	162	3
43	42	163	5
55	42	164	2
63	42	165	8
34	42	166	2
34	89	167	2
34	86	168	7
34	66	169	9
34	48	170	2
63	48	171	6
62	48	172	3
59	48	173	5
17	48	174	10
17	97	175	2
17	86	176	1
17	75	177	5
17	54	178	7
17	36	179	5
94	36	180	5
62	36	181	8
56	36	182	7
47	36	183	8
47	95	184	4
47	92	185	9
47	83	186	9
47	65	187	3
47	58	188	5
47	54	189	1
47	53	190	10
47	7	191	1
94	7	192	10
72	7	193	5
68	7	194	2

55	7	195	3
46	7	196	8
39	7	197	10
38	7	198	3
38	96	199	3
38	90	200	3
38	83	201	7
38	65	202	5
38	42	203	8
38	40	204	5
98	40	205	5
87	40	206	5
63	40	207	5
43	40	208	7
41	40	209	10
41	95	210	6
41	89	211	8
41	78	212	5
41	74	213	1
41	50	214	10
98	50	215	4
94	50	216	5
82	50	217	9
79	50	218	8
77	50	219	5
76	50	220	6
63	50	221	7
51	50	222	8
72	50	223	2
31	50	224	6
31	96	225	9
31	95	226	2
31	84	227	6
31	81	228	8
31	67	229	5
31	40	230	9
31	74	231	9
31	78	232	1
31	23	233	2
31	90	234	8
31	16	235	7
98	16	236	2
88	16	237	2
87	16	238	1
79	16	239	10
69	16	240	2
62	16	241	4
34	16	242	4
61	16	243	8
72	16	244	5
12	16	245	6
12	96	246	1
12	67	247	4

12	36	248	5
12	32	249	9
98	32	250	6
72	32	251	5
63	32	252	3
51	32	253	9
44	32	254	4
44	97	255	9
44	90	256	1
44	83	257	8
44	74	258	6
44	67	259	7
44	64	260	3
44	53	261	2
44	80	262	1
44	99	263	4
44	9	264	9
44	86	265	3
44	11	266	10
56	11	267	7
13	11	268	10
13	64	269	9
13	60	270	6
100	60	271	8
30	60	272	2
76	60	273	4
93	60	274	3
88	60	275	2
70	60	276	4
59	60	277	8
34	60	278	2
28	60	279	10
28	91	280	6
28	66	281	3
28	35	282	7
28	33	283	6
28	97	284	1
28	67	285	7
28	64	286	10
28	53	287	6
28	14	288	9
100	14	289	6
88	14	290	10
57	14	291	1
49	14	292	5
17	14	293	1
3	14	294	1
79	14	295	9
24	14	296	9
24	97	297	3
24	90	298	1
24	80	299	8
24	75	300	2

24	71	301	6
24	36	302	8
24	33	303	6
24	26	304	7
76	26	305	4
63	26	306	9
61	26	307	8
46	26	308	1
41	26	309	4
34	26	310	9
70	26	311	3
69	26	312	10
43	26	313	5
17	26	314	1
57	26	315	3
24	7	316	1
24	58	317	10
24	96	318	2
24	53	319	6
24	65	320	10
24	42	321	3
24	54	322	10
24	11	323	9
24	10	324	4
76	10	325	6
69	10	326	5
62	10	327	5
61	10	328	4
45	10	329	2
45	86	330	5
45	84	331	1
45	81	332	9
45	80	333	9
45	50	334	3
45	23	335	10
45	18	336	10
87	18	337	2
85	18	338	1
72	18	339	7
55	18	340	8
41	18	341	5
38	18	342	3
39	18	343	7
59	18	344	3
69	18	345	9
77	18	346	3
28	18	347	7
52	18	348	3
52	58	349	2
52	21	350	2
98	21	351	5
88	21	352	3
87	21	353	2

63	21	354	6
94	21	355	3
72	21	356	3
76	21	357	6
69	21	358	8
70	21	359	6
3	21	360	7
1	21	361	7
52	9	362	2
52	89	363	9
52	42	364	8
52	35	365	5
52	20	366	7
77	20	367	6
51	20	368	4
46	20	369	8
1	20	370	3
55	20	371	8
59	20	372	5
61	20	373	10
88	20	374	10
3	20	375	9
94	20	376	4
56	20	377	5
43	20	378	3
52	7	379	6
45	15	380	4
87	15	381	9
85	15	382	4
51	15	383	4
47	15	384	10
46	15	385	6
38	15	386	3
28	15	387	3
55	15	388	4
63	15	389	7
79	15	390	2
49	15	391	2
45	5	392	3
93	5	393	8
76	5	394	5
70	5	395	5
63	5	396	1
39	5	397	10
34	5	398	5
62	5	399	4
28	5	400	2
98	5	401	9
88	5	402	5
45	67	403	9
45	35	404	1
45	27	405	1
94	27	406	8

88	27	407	8
82	27	408	6
79	27	409	5
62	27	410	2
57	27	411	6
56	27	412	4
46	27	413	10
61	27	414	8
25	27	415	5
1	27	416	7
55	27	417	4
28	27	418	10
17	27	419	5
45	19	420	9
69	19	421	4
62	19	422	6
59	19	423	8
41	19	424	3
46	19	425	4
56	19	426	2
3	19	427	10
39	19	428	6
94	19	429	6
28	19	430	7
38	10	431	4
13	10	432	1
49	10	433	3
30	10	434	4
72	10	435	5
39	10	436	6
88	10	437	5
68	10	438	5
56	10	439	9
57	10	440	1
82	10	441	2
12	10	442	10
82	14	443	1
56	14	444	2
46	14	445	6
34	14	446	3
85	14	447	7
68	14	448	3
59	14	449	5
47	14	450	4
13	14	451	6
28	4	452	5
100	4	453	5
94	4	454	2
93	4	455	8
62	4	456	8
61	4	457	10
88	4	458	8
47	4	459	1



77	4	460	2
28	71	461	8
28	48	462	1
28	54	463	10
28	9	464	5
28	8	465	3
28	6	466	9
87	6	467	7
61	6	468	5
51	6	469	5
98	6	470	7
100	6	471	1
46	6	472	2
79	6	473	2
44	6	474	2
12	6	475	9
55	6	476	9
59	6	477	9
25	6	478	7
57	6	479	1
62	6	480	3
1	6	481	8
22	60	482	7
22	96	483	10
22	90	484	3
22	86	485	6
22	83	486	4
22	81	487	2
22	75	488	2
22	65	489	7
22	58	490	10
22	35	491	6
22	32	492	10
22	29	493	6
22	92	494	10
22	40	495	9
22	89	496	10
22	95	497	3
22	7	498	2
12	60	499	10
3	60	500	1
13	48	501	8
13	35	502	6
13	9	503	9
13	91	504	1
13	7	505	3
13	83	506	3
13	2	507	7
63	2	508	3
59	2	509	8
41	2	510	8
12	2	511	2
3	2	512	1

62	2	513	4
87	2	514	9
44	2	515	2
68	2	516	1
76	2	517	10
98	11	518	8
61	11	519	4
41	11	520	4
72	11	521	8
55	11	522	7
59	11	523	7
62	11	524	8
57	11	525	1
94	11	526	10
1	11	527	3
39	32	528	3
59	32	529	7
87	32	530	6
88	32	531	10
25	32	532	3
1	32	533	3
12	7	534	2
12	86	535	8
12	50	536	6
12	99	537	5
39	50	538	3
61	50	539	10
43	50	540	5
30	50	541	5
41	29	542	8
41	23	543	7
41	83	544	7
41	65	545	5
41	53	546	2
41	54	547	8
59	40	548	7
56	40	549	6
77	40	550	4
62	40	551	8
93	40	552	6
49	40	553	9
34	40	554	8
30	40	555	9
38	80	556	5
38	9	557	9
100	7	558	1
87	7	559	4
25	7	560	6
1	7	561	1
47	81	562	10
47	75	563	5
47	33	564	5
47	37	565	8

98	37	566	2
63	37	567	10
61	37	568	4
39	37	569	4
72	37	570	3
55	37	571	8
93	37	572	6
88	37	573	7
34	37	574	2
47	86	575	3
47	90	576	9
47	99	577	10
47	35	578	8
47	8	579	4
39	36	580	7
76	36	581	5
98	36	582	7
100	36	583	5
30	36	584	10
43	36	585	9
1	36	586	6
17	33	587	8
17	23	588	6
17	9	589	7
17	58	590	4
17	71	591	5
17	42	592	5
87	48	593	1
46	48	594	4
1	48	595	5
98	48	596	7
77	48	597	7
56	48	598	9
30	48	599	1
34	35	600	9
34	95	601	7
34	81	602	9
34	75	603	6
34	96	604	9
34	8	605	3
34	58	606	7
34	23	607	10
34	33	608	4
34	74	609	9
34	64	610	7
34	92	611	5
69	42	612	10
77	42	613	10
25	35	614	8
25	71	615	1
25	83	616	3
25	64	617	8
56	74	618	2

93	74	619	5
69	74	620	8
63	74	621	1
62	74	622	5
61	74	623	4
46	64	624	10
46	53	625	6
46	65	626	8
46	91	627	7
46	97	628	7
46	80	629	3
46	35	630	6
46	75	631	10
46	9	632	8
88	86	633	2
76	67	634	1
76	83	635	5
76	75	636	9
76	65	637	6
76	53	638	8
76	8	639	2
70	58	640	8
63	58	641	6
62	58	642	8
59	58	643	9
100	58	644	9
55	58	645	6
68	58	646	6
56	58	647	6
51	64	648	5
69	66	649	7
61	66	650	6
62	66	651	4
1	75	652	3
1	9	653	8
1	92	654	7
1	78	655	8
55	71	656	8
61	71	657	5
98	71	658	1
3	71	659	4
57	71	660	8
39	64	661	2
39	35	662	1
39	9	663	9
79	80	664	9
63	80	665	1
43	80	666	3
61	96	667	3
61	92	668	6
61	91	669	7
61	54	670	5
61	35	671	8

61	33	672	6
61	8	673	2
87	78	674	2
93	78	675	9
63	78	676	6
57	78	677	3
49	78	678	2
3	78	679	1
56	54	680	8
56	89	681	2
56	99	682	10
56	64	683	2
56	92	684	10
56	95	685	1
56	67	686	10
56	23	687	2
49	35	688	8
98	35	689	2
69	35	690	6
30	35	691	9
59	35	692	7
70	35	693	10
68	23	694	7
68	67	695	3
68	54	696	3
68	83	697	5
68	92	698	2
68	9	699	4
63	8	700	10
62	8	701	2
55	8	702	9
87	8	703	9
85	8	704	2
77	54	705	8
77	91	706	4
77	9	707	2
79	53	708	1
69	53	709	9
63	53	710	9
62	53	711	6
30	53	712	9
49	92	713	9
49	83	714	2
49	23	715	2
43	29	716	5
30	29	717	6
85	29	718	3
82	29	719	3
100	29	720	10
87	29	721	3
79	29	722	2
88	29	723	7
70	29	724	5

72	95	725	1
43	64	726	7
100	64	727	3
69	64	728	3
62	64	729	9
55	64	730	8
30	64	731	3
59	90	732	2
59	67	733	4
59	33	734	6
55	9	735	7
30	9	736	7
100	9	737	5
88	9	738	6
82	9	739	10
57	90	740	4
57	54	741	3
55	33	742	4
94	33	743	1
87	95	744	7
87	54	745	2
85	81	746	5
94	81	747	10
98	81	748	5
69	81	749	2
79	90	750	1
94	67	751	10
70	67	752	10
43	67	753	7
62	67	754	1
3	67	755	3
82	65	756	4
82	89	757	10
82	83	758	7
82	95	759	9
82	75	760	5
82	23	761	9
62	54	762	10
88	75	763	2
62	75	764	9
30	65	765	3
30	84	766	5
30	96	767	8
30	89	768	7
88	97	769	3
63	97	770	6
62	97	771	8
43	97	772	1
55	92	773	5
55	95	774	9
55	91	775	5
98	96	776	7
93	92	777	9

88	91	778	8
69	91	779	7
43	91	780	8
3	65	781	8
63	90	782	2
43	95	783	2
43	23	784	4
70	92	785	6
63	92	786	9
69	95	787	4
69	65	788	6
62	23	789	5
94	95	790	1
98	89	791	4
70	89	792	6
63	65	793	1
70	95	794	9
70	65	795	7
88	83	796	9
98	83	797	6
62	95	798	6
100	99	799	3

is valid partitioning? 1

## Aufgabe 2

Anmerkung: Für Graph3 liegt keine Bipartition vor und wurde daher nicht verwendet.

### Graph1

Executing BipartiteMatchingAlgorithm

```
{(1 -> 6)(3 -> 29)(7 -> 60)(8 -> 64)(9 -> 58)(11 -> 56)(12 -> 17)(14 -> 57)(16 -> 63)(20 -> 26)(22 ->
13)(25 -> 2)(27 -> 35)(28 -> 5)(31 -> 19)(32 -> 69)(38 -> 30)(41 -> 68)(43 ->
39)(44 -> 23)(45 -> 34)(47 -> 36)(48 -> 51)(49 -> 37)(50 -> 46)(55 -> 10)(59 -> 40)(61 -> 24)(66 ->
65)(67 -> 42)}
```

### Graph2

Executing BipartiteMatchingAlgorithm

```
{(1 -> 3)(4 -> 40)(5 -> 10)(6 -> 41)(11 -> 29)(12 -> 15)(13 -> 23)(14 -> 9)(16 -> 32)(24 -> 38)(25 ->
43)(26 -> 17)(28 -> 18)(31 -> 19)(33 -> 27)(34 -> 20)(35 -> 30)(36 -> 8)(37 -> 45)(39 -> 2)(42 -> 22)(46 ->
49)(47 -> 21)(50 -> 48)}
```

### Graph4

Executing BipartiteMatchingAlgorithm

```
{(1 -> 97)(3 -> 19)(12 -> 10)(13 -> 11)(17 -> 48)(22 -> 58)(24 -> 65)(25 -> 35)(28 -> 60)(30 -> 36)(31 ->
74)(34 -> 23)(38 -> 9)(39 -> 5)(41 -> 50)(43 -> 91)(44 -> 83)(45 -> 18)(46 -> 27)(47 -> 53)(49 -> 40)(51 ->
32)(52 -> 89)(55 -> 99)(56 -> 92)(57 -> 33)(59 -> 6)(61 -> 4)(62 -> 54)(63 -> 37)(68 -> 73)(69 ->
26)(70 -> 67)(72 -> 64)(76 -> 2)(77 -> 42)(79 -> 16)(82 -> 95)(85 -> 96)(87 -> 15)(88 -> 20)(93 -> 78)(94 ->
7)(98 -> 75)(100 -> 29)}
```

## Aufgabe 3

a)

Das Tippspiel-Problem kann als Suche eines Max. gewichteten Matchings in einem Bipartiten Graphen formuliert werden. Dabei ist die eine Partition die Menge der Spieltage und die andere die Menge der Fußballvereine. Es wird jeder Spieltag mit jedem Verein verbunden und erhält das Kantengewicht 3, falls die Mannschaft am jeweiligen Spieltag gewonnen hat, 1 bei einem Unentschieden und 0 bei einer Niederlage.

b)

Executing FootballBettingGameAlgorithm

```
{(name: 1.FCNürnberg -> number: 3 matches: {((name: BorussiaDortmund, name:
Bayer04Leverkusen) -> (3, 0)) ((name: EintrachtFrankfurt, name: HamburgerSV) -> (3, 2)) ((name:
FCAugsburg, name: VfLWolfsburg) -> (0, 0)) ((name: FCBayernMünchen, name: 1.FSVMainz05) -> (3,
1)) ((name: Hannover96, name: SVWerderBremen) -> (3, 2)) ((name: M'gladbach, name:
1.FCNürnberg) -> (2, 3)) ((name: SpVggGreutherFürth, name: FCSchalke04) -> (0, 2)) ((name: Sport-
ClubFreiburg, name: 1899Hoffenheim) -> (5, 3)) ((name: VfBStuttgart, name: FortunaDüsseldorf) ->
(0, 0)) })
```

```
(name: 1.FSVMainz05 -> number: 9 matches: {((name: 1.FSVMainz05, name: 1899Hoffenheim) ->
(3, 0)) ((name: FCAugsburg, name: HamburgerSV) -> (0, 2)) ((name: FCBayernMünchen,
name: Bayer04Leverkusen) -> (1, 2)) ((name: FCSchalke04, name: 1.FCNürnberg) -> (1, 0)) ((name:
FortunaDüsseldorf, name: VfLWolfsburg) -> (1, 4)) ((name: Hannover96, name: M'gladbach) -> (2, 3))
((name: SpVggGreutherFürth, name: SVWerderBremen) -> (1, 1)) ((name: Sport-ClubFreiburg, name:
BorussiaDortmund) -> (0, 2)) ((name: VfBStuttgart, name: EintrachtFrankfurt) -> (2, 1)) })
```

```
(name: 1899Hoffenheim -> number: 4 matches: {((name: 1.FCNürnberg, name: EintrachtFrankfurt)
-> (1, 2)) ((name: 1.FSVMainz05, name: FCAugsburg) -> (2, 0)) ((name: 1899Hoffenheim, name:
Hannover96) -> (3, 1)) ((name: Bayer04Leverkusen, name: M'gladbach) -> (1, 1)) ((name:
FCSchalke04, name: FCBayernMünchen) -> (0, 2)) ((name: FortunaDüsseldorf, name: Sport-
ClubFreiburg) -> (0, 0)) ((name: HamburgerSV, name: BorussiaDortmund) -> (3, 2)) ((name:
SVWerderBremen, name: VfBStuttgart) -> (2, 2)) ((name: VfLWolfsburg, name: SpVggGreutherFürth)
-> (1, 1)) })
```

```
(name: Bayer04Leverkusen -> number: 5 matches: {((name: EintrachtFrankfurt, name:
BorussiaDortmund) -> (3, 3)) ((name: FCAugsburg, name: Bayer04Leverkusen) -> (1, 3)) ((name:
FCBayernMünchen, name: VfLWolfsburg) -> (3, 0)) ((name: FCSchalke04, name: 1.FSVMainz05) -> (3,
0)) ((name: Hannover96, name: 1.FCNürnberg) -> (4, 1)) ((name: M'gladbach, name: HamburgerSV) -
> (2, 2)) ((name: SpVggGreutherFürth, name: FortunaDüsseldorf) -> (0, 2)) ((name: Sport-
ClubFreiburg, name: SVWerderBremen) -> (1, 2)) ((name: VfBStuttgart, name: 1899Hoffenheim) ->
(0, 3)) })
```

```
(name: BorussiaDortmund -> number: 13 matches: {((name: 1.FSVMainz05, name:
BorussiaDortmund) -> (1, 2)) ((name: 1899Hoffenheim, name: Bayer04Leverkusen) -> (1, 2)) ((name:
FCAugsburg, name: M'gladbach) -> (1, 1)) ((name: FCBayernMünchen, name: Hannover96) -> (5, 0))
((name: FCSchalke04, name: EintrachtFrankfurt) -> (1, 1)) ((name: FortunaDüsseldorf, name:
HamburgerSV) -> (2, 0)) ((name: SpVggGreutherFürth, name: 1.FCNürnberg) -> (0, 0)) ((name: Sport-
ClubFreiburg, name: VfBStuttgart) -> (3, 0)) ((name: VfLWolfsburg, name: SVWerderBremen) -> (1,
1)) })
```

```
(name: EintrachtFrankfurt -> number: 16 matches: {((name: 1.FCNürnberg, name:
FortunaDüsseldorf) -> (2, 0)) ((name: BorussiaDortmund, name: VfLWolfsburg) -> (2, 3)) ((name:
```



EintrachtFrankfurt, name: SVWerderBremen) -> (4, 1)) ((name: FCAugsburg, name: FCBayernMünchen) -> (0, 2)) ((name: HamburgerSV, name: 1899Hoffenheim) -> (2, 0)) ((name: Hannover96, name: Bayer04Leverkusen) -> (3, 2)) ((name: M'gladbach, name: 1.FSVMainz05) -> (2, 0)) ((name: Sport-ClubFreiburg, name: SpVggGreutherFürth) -> (1, 0)) ((name: VfBStuttgart, name: FCSchalke04) -> (3, 1)) }}

(name: FCAugsburg -> number: 7 matches: (((name: 1.FSVMainz05, name: FortunaDüsseldorf) -> (1, 0)) ((name: FCAugsburg, name: SVWerderBremen) -> (3, 1)) ((name: FCBayernMünchen, name: 1899Hoffenheim) -> (2, 0)) ((name: FCSchalke04, name: VfLWolfsburg) -> (3, 0)) ((name: Hannover96, name: BorussiaDortmund) -> (1, 1)) ((name: M'gladbach, name: EintrachtFrankfurt) -> (2, 0)) ((name: SpVggGreutherFürth, name: HamburgerSV) -> (0, 1)) ((name: Sport-ClubFreiburg, name: 1.FCNürnberg) -> (3, 0)) ((name: VfBStuttgart, name: Bayer04Leverkusen) -> (2, 2)) ))

(name: FCBayernMünchen -> number: 10 matches: (((name: 1.FCNürnberg, name: VfLWolfsburg) -> (1, 0)) ((name: 1899Hoffenheim, name: FCSchalke04) -> (3, 2)) ((name: Bayer04Leverkusen, name: FortunaDüsseldorf) -> (3, 2)) ((name: BorussiaDortmund, name: VfBStuttgart) -> (0, 0)) ((name: EintrachtFrankfurt, name: SpVggGreutherFürth) -> (1, 1)) ((name: HamburgerSV, name: FCBayernMünchen) -> (0, 3)) ((name: Hannover96, name: FCAugsburg) -> (2, 0)) ((name: M'gladbach, name: Sport-ClubFreiburg) -> (1, 1)) ((name: SVWerderBremen, name: 1.FSVMainz05) -> (2, 1)) ))

(name: FCSchalke04 -> number: 11 matches: (((name: 1.FSVMainz05, name: 1.FCNürnberg) -> (2, 1)) ((name: FCAugsburg, name: BorussiaDortmund) -> (1, 3)) ((name: FCBayernMünchen, name: EintrachtFrankfurt) -> (2, 0)) ((name: FCSchalke04, name: SVWerderBremen) -> (2, 1)) ((name: FortunaDüsseldorf, name: 1899Hoffenheim) -> (1, 1)) ((name: SpVggGreutherFürth, name: M'gladbach) -> (2, 4)) ((name: Sport-ClubFreiburg, name: HamburgerSV) -> (0, 0)) ((name: VfBStuttgart, name: Hannover96) -> (2, 4)) ((name: VfLWolfsburg, name: Bayer04Leverkusen) -> (3, 1)) ))

(name: FortunaDüsseldorf -> number: 15 matches: (((name: 1.FSVMainz05, name: Hannover96) -> (2, 1)) ((name: 1899Hoffenheim, name: SVWerderBremen) -> (1, 4)) ((name: Bayer04Leverkusen, name: 1.FCNürnberg) -> (1, 0)) ((name: FCAugsburg, name: Sport-ClubFreiburg) -> (1, 1)) ((name: FCBayernMünchen, name: BorussiaDortmund) -> (1, 1)) ((name: FCSchalke04, name: M'gladbach) -> (1, 1)) ((name: FortunaDüsseldorf, name: EintrachtFrankfurt) -> (4, 0)) ((name: SpVggGreutherFürth, name: VfBStuttgart) -> (0, 1)) ((name: VfLWolfsburg, name: HamburgerSV) -> (1, 1)) ))

(name: HamburgerSV -> number: 12 matches: (((name: 1.FCNürnberg, name: FCBayernMünchen) -> (1, 1)) ((name: 1899Hoffenheim, name: VfLWolfsburg) -> (1, 3)) ((name: Bayer04Leverkusen, name: FCSchalke04) -> (2, 0)) ((name: BorussiaDortmund, name: SpVggGreutherFürth) -> (3, 1)) ((name: EintrachtFrankfurt, name: FCAugsburg) -> (4, 2)) ((name: HamburgerSV, name: 1.FSVMainz05) -> (1, 0)) ((name: Hannover96, name: Sport-ClubFreiburg) -> (1, 2)) ((name: M'gladbach, name: VfBStuttgart) -> (1, 2)) ((name: SVWerderBremen, name: FortunaDüsseldorf) -> (2, 1)) ))

(name: Hannover96 -> number: 14 matches: (((name: 1.FCNürnberg, name: 1899Hoffenheim) -> (4, 2)) ((name: BorussiaDortmund, name: FortunaDüsseldorf) -> (1, 1)) ((name: EintrachtFrankfurt, name: 1.FSVMainz05) -> (1, 3)) ((name: HamburgerSV, name: FCSchalke04) -> (3, 1)) ((name: Hannover96, name: SpVggGreutherFürth) -> (2, 0)) ((name: M'gladbach, name: VfLWolfsburg) -> (2, 0)) ((name: SVWerderBremen, name: Bayer04Leverkusen) -> (1, 4)) ((name: Sport-ClubFreiburg, name: FCBayernMünchen) -> (0, 2)) ((name: VfBStuttgart, name: FCAugsburg) -> (2, 1)) ))

```
(name: M'gladbach -> number: 1 matches: {((name: BorussiaDortmund, name: SVWerderBremen)
-> (2, 1)) ((name: EintrachtFrankfurt, name: Bayer04Leverkusen) -> (2, 1)) ((name: FCAugsburg,
name: FortunaDüsseldorf) -> (0, 2)) ((name: HamburgerSV, name: 1.FCNürnberg) -> (0, 1)) ((name:
Hannover96, name: FCSchalke04) -> (2, 2)) ((name: M'gladbach, name: 1899Hoffenheim) -> (2, 1))
((name: SpVggGreutherFürth, name: FCBayernMünchen) -> (0, 3)) ((name: Sport-ClubFreiburg,
name: 1.FSVMainz05) -> (1, 1)) ((name: VfBStuttgart, name: VfLWolfsburg) -> (0, 1)) })
```

```
(name: SVWerderBremen -> number: 8 matches: {((name: 1.FCNürnberg, name: FCAugsburg) ->
(0, 0)) ((name: 1899Hoffenheim, name: SpVggGreutherFürth) -> (3, 3)) ((name: Bayer04Leverkusen,
name: 1.FSVMainz05) -> (2, 2)) ((name: BorussiaDortmund, name: FCSchalke04) -> (1, 2)) ((name:
EintrachtFrankfurt, name: Hannover96) -> (3, 1)) ((name: FortunaDüsseldorf, name:
FCBayernMünchen) -> (0, 5)) ((name: HamburgerSV, name: VfBStuttgart) -> (0, 1)) ((name:
SVWerderBremen, name: M'gladbach) -> (4, 0)) ((name: VfLWolfsburg, name: Sport-ClubFreiburg) ->
(0, 2)) })
```

```
(name: SpVggGreutherFürth -> number: 2 matches: {((name: 1.FCNürnberg, name:
BorussiaDortmund) -> (1, 1)) ((name: 1.FSVMainz05, name: SpVggGreutherFürth) -> (0, 1)) ((name:
1899Hoffenheim, name: EintrachtFrankfurt) -> (0, 4)) ((name: Bayer04Leverkusen, name: Sport-
ClubFreiburg) -> (2, 0)) ((name: FCBayernMünchen, name: VfBStuttgart) -> (6, 1)) ((name:
FCSchalke04, name: FCAugsburg) -> (3, 1)) ((name: FortunaDüsseldorf, name: M'gladbach) -> (0, 0))
((name: SVWerderBremen, name: HamburgerSV) -> (2, 0)) ((name: VfLWolfsburg, name:
Hannover96) -> (0, 4)) })
```

```
(name: Sport-ClubFreiburg -> number: 17 matches: {((name: 1.FSVMainz05, name: VfBStuttgart) ->
(3, 1)) ((name: 1899Hoffenheim, name: BorussiaDortmund) -> (1, 3)) ((name: Bayer04Leverkusen,
name: HamburgerSV) -> (3, 0)) ((name: FCBayernMünchen, name: M'gladbach) -> (1, 1)) ((name:
FCSchalke04, name: Sport-ClubFreiburg) -> (1, 3)) ((name: FortunaDüsseldorf, name: Hannover96) ->
(2, 1)) ((name: SVWerderBremen, name: 1.FCNürnberg) -> (1, 1)) ((name: SpVggGreutherFürth,
name: FCAugsburg) -> (1, 1)) ((name: VfLWolfsburg, name:
EintrachtFrankfurt) -> (0, 2)) })
```

```
(name: VfBStuttgart -> number: 6 matches: {((name: 1.FCNürnberg, name: VfBStuttgart) -> (0, 2))
((name: 1899Hoffenheim, name: FCAugsburg) -> (0, 0)) ((name: Bayer04Leverkusen, name:
SpVggGreutherFürth) -> (2, 0)) ((name: BorussiaDortmund, name: M'gladbach) -> (5, 0)) ((name:
EintrachtFrankfurt, name: Sport-ClubFreiburg) -> (2, 1)) ((name: FortunaDüsseldorf, name:
FCSchalke04) -> (2, 2)) ((name: HamburgerSV, name: Hannover96) -> (1, 0)) ((name:
SVWerderBremen, name: FCBayernMünchen) -> (0, 2)) ((name: VfLWolfsburg, name: 1.FSVMainz05)
-> (0, 2)) })
```

```
}
```

## Aufgabe 4

Anmerkung: Das TSP Problem auf  $f(G)$  wird nicht gelöst, wenn  $G$  aus mehreren Zusammenhangskomponenten besteht, da  $f(G)$  dann nicht vollständig ist. In diesem Fall wird ein Graph bestehend aus den Knoten einer beliebigen Zusammenhangskomponente ausgegeben.

### Graph1

Executing ChristofidesAlgorithm

@nodes

label

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  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70

@edges

		label	weight
69	67	123	1
68	6	127	2
67	40	236	1
66	60	287	2
65	21	351	2
64	55	405	2
63	48	460	2
61	19	570	1
59	4	632	2
58	21	728	1
58	4	730	1
57	31	783	1
56	12	827	2
56	14	828	2
55	63	884	2
54	32	909	5
52	30	1022	4
51	22	1067	1
50	36	1116	2
50	26	1120	2
49	23	1162	1
48	57	1211	3
47	69	1250	1
44	41	1367	5
42	14	1448	1
39	62	1556	2
39	12	1559	3
38	5	1598	1
37	61	1637	1
37	31	1640	2
35	47	1707	2
34	3	1742	2
34	8	1745	1
30	3	1865	6

29	44	1903	3
27	26	1958	4
25	42	2001	1
25	46	2002	1
24	70	2026	1
24	15	2027	1
23	54	2056	1
19	20	2146	1
18	35	2171	1
18	51	2174	1
17	28	2188	1
16	60	2213	2
16	64	2215	1
15	46	2231	2
13	53	2271	3
11	36	2303	1
10	45	2317	2
6	11	2371	1
5	2	2404	4
41	29	2405	3
9	29	2406	6
45	9	2407	4
43	1	2408	3
2	27	2409	1
40	43	2410	4
59	58	2411	2
7	33	2412	5
2	66	2413	1
13	7	2414	5
32	64	2415	5
33	64	2416	2
28	33	2417	2
68	17	2418	2
1	10	2419	1
62	20	2420	3

@nodes

label

@edges

Label

Graph2

Executing ChristofidesAlgorithm

@nodes

label

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  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50

@edges

		label	weight
50	18	40	3
50	9	47	2
48	26	136	1
47	19	174	4
47	32	177	2
46	8	210	1
44	11	295	1

41	39	412	5
40	25	445	1
39	45	482	1
38	20	506	8
37	43	551	3
36	24	573	4
35	21	616	1
34	40	651	1
33	3	678	1
32	13	707	1
31	2	738	1
30	4	765	1
28	38	816	3
27	13	840	2
26	45	867	2
25	49	890	1
24	49	915	1
23	46	940	1
22	8	946	5
21	42	982	2
20	12	1001	2
19	33	1020	4
18	5	1036	1
16	10	1068	1
15	37	1077	2
15	28	1084	2
14	30	1095	1
11	10	1134	1
9	12	1161	1
7	35	1178	3
6	17	1186	2
5	29	1198	3
2	14	1218	1
31	29	1222	4
6	17	1223	3
27	17	1224	2
23	46	1225	3
34	11	1226	4
42	36	1227	3
16	7	1228	4
48	26	1229	3
1	41	1230	3
43	1	1231	9

@nodes

label

@edges

label

Graph3

Executing ChristofidesAlgorithm

```
@nodes
label
@edges
    label weight
```

```
@nodes
label
250
@edges
    label
```

## Graph4

```
Executing ChristofidesAlgorithm
@nodes
label
@edges
    label weight
```

```
@nodes
label
150
@edges
    label
```