**Denis Baskan – 878 571**

**Yasin Said – 862471**

**Sah Ranjit – 836261**

**Datum: 28.06.2019**

**Fach: Operations Research**

**Hausaufgabe2**

**Aufgabe 1 Das Lawler Verfahren**

Um eine optimale Zuordnung von S = V nach T = V zu bekommen, programmierten wir den Lawler Algorithmus in Python, wobei die Knotennummerierung (0,…,7) statt (1,…,8) ist. Somit werden die Knoten aus S mit {s0,...,s7} und die Knoten aus T mit {t0,...,t7} bezeichnet.

Kurzbeschreibung der Variablen:

* I – Iterationsschritt
* w – Gewichtsmatrix
* x – Zuordnungsmatrix
* u, v – Variablen des dualen, linearen Programms.
* ER – Rote Kanten, die zu zugeordnet sind
* EB – Blaue Kanten, die noch nicht zugeordneten sind
* c – Kantengewichte des Hilfsgraphs G
* G – Gerichteter Hilfsgraph
* J – Knoten, die noch nicht zugeordnet wurden
* ustrich, vstrich – Wegentfernungen
* k – Index zum Knoten mit kürzestem Weg
* vk – Kante zum Knoten mit kürzestem Weg
* P – Kürzeste Weg

**Ergebnisse:**

**Initialisierung**

w:

[[1000 3 4 5 6 7 8 9]

[ 3 1000 9 8 7 6 5 4]

[ 4 9 1000 8 7 6 5 9]

[ 5 8 8 1000 4 5 6 9]

[ 6 7 7 4 1000 8 2 8]

[ 7 6 6 5 8 1000 4 3]

[ 8 5 5 6 2 4 1000 8]

[ 9 4 9 9 8 3 8 1000]]

x:

[[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]]

u,v,t: [0 0 0 0 0 0 0 0], [0 0 0 0 0 0 0 0], 1

x:

[[0 1 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]]

u,v,t: [0 3 3 3 3 3 3 3], [0 3 0 0 0 0 0 0], 1

######################## Nächste Iteration ########################

ER: [(0, 1)]

EB: [(0, 0), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (7, 0), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7)]

c:

[[1000 0 4 5 6 7 8 9]

[ 6 1000 12 11 10 9 8 7]

[ 7 9 1003 11 10 9 8 12]

[ 8 8 11 1003 7 8 9 12]

[ 9 7 10 7 1003 11 5 11]

[ 10 6 9 8 11 1003 7 6]

[ 11 5 8 9 5 7 1003 11]

[ 12 4 12 12 11 6 11 1003]]

Gewichte in G:

{('s5', 't5'): 1003, ('s0', 't5'): 7, ('s6', 't5'): 7, ('s6', 't3'): 9, ('s0', 't0'): 1000, ('s0', 't6'): 8, ('s6', 't7'): 11, ('s1', 't3'): 11, ('s3', 't7'): 12, ('s3', 't1'): 8, ('s7', 't6'): 11, ('s1', 't2'): 12, ('s5', 't3'): 8, ('s2', 't1'): 9, ('s2', 't7'): 12, ('t1', 's0'): 0, ('s4', 't2'): 10, ('s4', 't5'): 11, ('s4', 't6'): 5, ('s2', 't3'): 11, ('s5', 't1'): 6, ('s3', 't4'): 7, ('s4', 't0'): 9, ('s3', 't2'): 11, ('s1', 't1'): 1000, ('s1', 't7'): 7, ('s6', 't4'): 5, ('s1', 't6'): 8, ('s0', 't3'): 5, ('s5', 't7'): 6, ('s7', 't3'): 12, ('s6', 't2'): 8, ('s5', 't6'): 7, ('s5', 't4'): 11, ('s6', 't6'): 1003, ('s3', 't5'): 8, ('s2', 't4'): 10, ('s3', 't3'): 1003, ('s4', 't4'): 1003, ('s7', 't2'): 12, ('s5', 't0'): 10, ('s6', 't1'): 5, ('s1', 't0'): 6, ('s7', 't4'): 11, ('s7', 't7'): 1003, ('s7', 't5'): 6, ('s6', 't0'): 11, ('s2', 't5'): 9, ('s2', 't0'): 7, ('s5', 't2'): 9, ('s3', 't6'): 9, ('s7', 't1'): 4, ('s0', 't2'): 4, ('s4', 't3'): 7, ('s2', 't6'): 8, ('s3', 't0'): 8, ('s2', 't2'): 1003, ('s0', 't7'): 9, ('s0', 't4'): 6, ('s4', 't7'): 11, ('s4', 't1'): 7, ('s7', 't0'): 12, ('s1', 't4'): 10, ('s1', 't5'): 9}

J: [2 3 4 5 6 7]

vstrich: [6 1000 12 11 10 9 8 7]

ustrich: [1000. 0. inf inf inf inf inf inf]

k,P,vk: 7, ['s1', 't7'], 7

u: [7 3 10 10 10 10 10 10]

v: [6 10 7 7 7 7 7 7]

x:

[[0 1 0 0 0 0 0 0]

[0 0 0 0 0 0 0 1]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]]

P: ['s1', 't7']

######################## Nächste Iteration ########################

ER: [(0, 1), (1, 7)]

EB: [(0, 0), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (7, 0), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7)]

c:

[[1001 0 4 5 6 7 8 9]

[ 0 993 5 4 3 2 1 0]

[ 8 9 1003 11 10 9 8 12]

[ 9 8 11 1003 7 8 9 12]

[ 10 7 10 7 1003 11 5 11]

[ 11 6 9 8 11 1003 7 6]

[ 12 5 8 9 5 7 1003 11]

[ 13 4 12 12 11 6 11 1003]]

Gewichte in G:

{('s5', 't5'): 1003, ('s0', 't5'): 7, ('s6', 't5'): 7, ('s6', 't3'): 9, ('s3', 't7'): 12, ('s0', 't6'): 8, ('s6', 't7'): 11, ('s1', 't3'): 4, ('s5', 't0'): 11, ('s3', 't1'): 8, ('s7', 't6'): 11, ('s1', 't2'): 5, ('s5', 't3'): 8, ('s2', 't1'): 9, ('s2', 't7'): 12, ('t1', 's0'): 0, ('s4', 't2'): 10, ('s4', 't5'): 11, ('s4', 't6'): 5, ('s2', 't3'): 11, ('s5', 't1'): 6, ('s3', 't4'): 7, ('s4', 't0'): 10, ('s3', 't2'): 11, ('s1', 't1'): 993, ('s6', 't4'): 5, ('s1', 't6'): 1, ('s0', 't3'): 5, ('s5', 't7'): 6, ('s7', 't3'): 12, ('s6', 't2'): 8, ('s5', 't6'): 7, ('s5', 't4'): 11, ('s6', 't6'): 1003, ('s3', 't5'): 8, ('s2', 't4'): 10, ('s3', 't3'): 1003, ('s4', 't4'): 1003, ('s7', 't2'): 12, ('t7', 's1'): 0, ('s0', 't0'): 1001, ('s6', 't1'): 5, ('s1', 't0'): 0, ('s7', 't4'): 11, ('s7', 't7'): 1003, ('s7', 't5'): 6, ('s6', 't0'): 12, ('s2', 't5'): 9, ('s2', 't0'): 8, ('s5', 't2'): 9, ('s3', 't6'): 9, ('s7', 't1'): 4, ('s0', 't2'): 4, ('s4', 't3'): 7, ('s2', 't6'): 8, ('s3', 't0'): 9, ('s2', 't2'): 1003, ('s0', 't7'): 9, ('s0', 't4'): 6, ('s4', 't7'): 11, ('s4', 't1'): 7, ('s7', 't0'): 13, ('s1', 't4'): 3, ('s1', 't5'): 2}

J: [2 3 4 5 6]

vstrich: [8 9 13 11 10 9 8 12]

ustrich: [9. 12. 0. inf inf inf inf inf]

k,P,vk: 6, ['s2', 't6'], 8

u: [15 11 10 18 18 18 18 18]

v: [14 18 15 15 15 15 15 15]

x:

[[0 1 0 0 0 0 0 0]

[0 0 0 0 0 0 0 1]

[0 0 0 0 0 0 1 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]]

P: ['s2', 't6']

######################## Nächste Iteration ########################

ER: [(0, 1), (1, 7), (2, 6)]

EB: [(0, 0), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (7, 0), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7)]

c:

[[1001 0 4 5 6 7 8 9]

[ 0 993 5 4 3 2 1 0]

[ 0 1 995 3 2 1 0 4]

[ 9 8 11 1003 7 8 9 12]

[ 10 7 10 7 1003 11 5 11]

[ 11 6 9 8 11 1003 7 6]

[ 12 5 8 9 5 7 1003 11]

[ 13 4 12 12 11 6 11 1003]]

Gewichte in G:

{('s5', 't5'): 1003, ('s0', 't5'): 7, ('s6', 't5'): 7, ('s6', 't3'): 9, ('s3', 't7'): 12, ('s0', 't6'): 8, ('s6', 't7'): 11, ('s1', 't3'): 4, ('s5', 't0'): 11, ('s3', 't1'): 8, ('s7', 't6'): 11, ('s1', 't2'): 5, ('s5', 't3'): 8, ('s2', 't1'): 1, ('s2', 't7'): 4, ('t1', 's0'): 0, ('s4', 't2'): 10, ('s4', 't5'): 11, ('s4', 't6'): 5, ('s2', 't3'): 3, ('s5', 't1'): 6, ('s3', 't4'): 7, ('s4', 't0'): 10, ('s3', 't2'): 11, ('s1', 't1'): 993, ('s6', 't4'): 5, ('s1', 't6'): 1, ('s0', 't3'): 5, ('t6', 's2'): 0, ('s5', 't7'): 6, ('s7', 't3'): 12, ('s6', 't2'): 8, ('s5', 't6'): 7, ('s5', 't4'): 11, ('s6', 't6'): 1003, ('s3', 't5'): 8, ('s2', 't4'): 2, ('s3', 't3'): 1003, ('s4', 't4'): 1003, ('s7', 't2'): 12, ('t7', 's1'): 0, ('s0', 't0'): 1001, ('s6', 't1'): 5, ('s1', 't0'): 0, ('s7', 't4'): 11, ('s7', 't7'): 1003, ('s7', 't5'): 6, ('s6', 't0'): 12, ('s2', 't5'): 1, ('s2', 't0'): 0, ('s5', 't2'): 9, ('s3', 't6'): 9, ('s7', 't1'): 4, ('s0', 't2'): 4, ('s4', 't3'): 7, ('s3', 't0'): 9, ('s2', 't2'): 995, ('s0', 't7'): 9, ('s0', 't4'): 6, ('s4', 't7'): 11, ('s4', 't1'): 7, ('s7', 't0'): 13, ('s1', 't4'): 3, ('s1', 't5'): 2}

J: [2 3 4 5]

vstrich: [ 9 8 11 12 7 8 9 12]

ustrich: [ 8. 12. 9. 0. inf inf inf inf]

k,P,vk: 4, ['s3', 't4'], 7

u: [22 18 17 18 25 25 25 25]

v: [21 25 22 22 22 22 22 22]

x:

[[0 1 0 0 0 0 0 0]

[0 0 0 0 0 0 0 1]

[0 0 0 0 0 0 1 0]

[0 0 0 0 1 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]]

P: ['s3', 't4']

######################## Nächste Iteration ########################

ER: [(0, 1), (1, 7), (2, 6), (3, 4)]

EB: [(0, 0), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 5), (3, 6), (3, 7), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (7, 0), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7)]

c:

[[1001 0 4 5 6 7 8 9]

[ 0 993 5 4 3 2 1 0]

[ 0 1 995 3 2 1 0 4]

[ 2 1 4 996 0 1 2 5]

[ 10 7 10 7 1003 11 5 11]

[ 11 6 9 8 11 1003 7 6]

[ 12 5 8 9 5 7 1003 11]

[ 13 4 12 12 11 6 11 1003]]

Gewichte in G:

{('s5', 't5'): 1003, ('s0', 't5'): 7, ('s6', 't5'): 7, ('s6', 't3'): 9, ('s3', 't7'): 5, ('t4', 's3'): 0, ('s0', 't6'): 8, ('s6', 't7'): 11, ('s1', 't3'): 4, ('s5', 't0'): 11, ('s3', 't1'): 1, ('s7', 't6'): 11, ('s1', 't2'): 5, ('s5', 't3'): 8, ('s2', 't1'): 1, ('s2', 't7'): 4, ('t1', 's0'): 0, ('s4', 't2'): 10, ('s4', 't5'): 11, ('s4', 't6'): 5, ('s2', 't3'): 3, ('s5', 't1'): 6, ('s4', 't0'): 10, ('s3', 't2'): 4, ('s1', 't1'): 993, ('s6', 't4'): 5, ('s1', 't6'): 1, ('s0', 't3'): 5, ('t6', 's2'): 0, ('s5', 't7'): 6, ('s7', 't3'): 12, ('s6', 't2'): 8, ('s5', 't6'): 7, ('s5', 't4'): 11, ('s6', 't6'): 1003, ('s3', 't5'): 1, ('s2', 't4'): 2, ('s3', 't3'): 996, ('s4', 't4'): 1003, ('s7', 't2'): 12, ('t7', 's1'): 0, ('s0', 't0'): 1001, ('s6', 't1'): 5, ('s1', 't0'): 0, ('s7', 't4'): 11, ('s7', 't7'): 1003, ('s7', 't5'): 6, ('s6', 't0'): 12, ('s2', 't5'): 1, ('s2', 't0'): 0, ('s5', 't2'): 9, ('s3', 't6'): 2, ('s7', 't1'): 4, ('s0', 't2'): 4, ('s4', 't3'): 7, ('s3', 't0'): 2, ('s2', 't2'): 995, ('s0', 't7'): 9, ('s0', 't4'): 6, ('s4', 't7'): 11, ('s4', 't1'): 7, ('s7', 't0'): 13, ('s1', 't4'): 3, ('s1', 't5'): 2}

J: [2 3 5]

vstrich: [ 5 6 10 7 7 6 5 9]

ustrich: [ 6. 9. 5. 7. 0. inf inf inf]

k,P,vk: 5, ['s4', 't6', 's2', 't5'], 6

u: [28 24 22 24 25 31 31 31]

v: [26 31 28 28 28 28 27 28]

x:

[[0 1 0 0 0 0 0 0]

[0 0 0 0 0 0 0 1]

[0 0 0 0 0 1 0 0]

[0 0 0 0 1 0 0 0]

[0 0 0 0 0 0 1 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]]

P: ['s4', 't6', 's2', 't5']

######################## Nächste Iteration ########################

ER: [(0, 1), (1, 7), (2, 5), (3, 4), (4, 6)]

EB: [(0, 0), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (2, 6), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 5), (3, 6), (3, 7), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 7), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (7, 0), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7)]

c:

[[1002 0 4 5 6 7 9 9]

[ 1 993 5 4 3 2 2 0]

[ 0 0 994 2 1 0 0 3]

[ 3 1 4 996 0 1 3 5]

[ 5 1 4 1 997 5 0 5]

[ 12 6 9 8 11 1003 8 6]

[ 13 5 8 9 5 7 1004 11]

[ 14 4 12 12 11 6 12 1003]]

Gewichte in G:

{('s5', 't5'): 1003, ('s0', 't5'): 7, ('t6', 's4'): 0, ('s6', 't5'): 7, ('s6', 't3'): 9, ('s3', 't7'): 5, ('t4', 's3'): 0, ('s0', 't6'): 9, ('s6', 't7'): 11, ('s1', 't3'): 4, ('s5', 't0'): 12, ('s3', 't1'): 1, ('s7', 't6'): 12, ('s1', 't2'): 5, ('s5', 't3'): 8, ('s2', 't1'): 0, ('s2', 't7'): 3, ('t1', 's0'): 0, ('s4', 't2'): 4, ('s4', 't5'): 5, ('t5', 's2'): 0, ('s2', 't3'): 2, ('s5', 't1'): 6, ('s4', 't0'): 5, ('s3', 't2'): 4, ('s1', 't1'): 993, ('s6', 't4'): 5, ('s1', 't6'): 2, ('s0', 't3'): 5, ('s5', 't7'): 6, ('s7', 't3'): 12, ('s6', 't2'): 8, ('s5', 't6'): 8, ('s5', 't4'): 11, ('s6', 't6'): 1004, ('s3', 't5'): 1, ('s2', 't4'): 1, ('s3', 't3'): 996, ('s4', 't4'): 997, ('s7', 't2'): 12, ('t7', 's1'): 0, ('s0', 't0'): 1002, ('s6', 't1'): 5, ('s1', 't0'): 1, ('s7', 't4'): 11, ('s7', 't7'): 1003, ('s7', 't5'): 6, ('s6', 't0'): 13, ('s2', 't0'): 0, ('s5', 't2'): 9, ('s3', 't6'): 3, ('s7', 't1'): 4, ('s0', 't2'): 4, ('s4', 't3'): 1, ('s2', 't6'): 0, ('s3', 't0'): 3, ('s2', 't2'): 994, ('s0', 't7'): 9, ('s0', 't4'): 6, ('s4', 't7'): 5, ('s4', 't1'): 1, ('s7', 't0'): 14, ('s1', 't4'): 3, ('s1', 't5'): 2}

J: [2 3]

vstrich: [7 6 9 8 9 8 8 6]

ustrich: [ 6. 6. 8. 9. 8. 0. inf inf]

k,P,vk: 3, ['s5', 't3'], 8

u: [34 30 30 32 33 31 39 39]

v: [33 37 36 36 36 36 35 34]

x:

[[0 1 0 0 0 0 0 0]

[0 0 0 0 0 0 0 1]

[0 0 0 0 0 1 0 0]

[0 0 0 0 1 0 0 0]

[0 0 0 0 0 0 1 0]

[0 0 0 1 0 0 0 0]

[0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0]]

P: ['s5', 't3']

######################## Nächste Iteration ########################

ER: [(0, 1), (1, 7), (2, 5), (3, 4), (4, 6), (5, 3)]

EB: [(0, 0), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (2, 6), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 5), (3, 6), (3, 7), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 7), (5, 0), (5, 1), (5, 2), (5, 4), (5, 5), (5, 6), (5, 7), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (7, 0), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7)]

c:

[[1001 0 2 3 4 5 7 9]

[ 0 993 3 2 1 0 0 0]

[ 1 2 994 2 1 0 0 5]

[ 4 3 4 996 0 1 3 7]

[ 6 3 4 1 997 5 0 7]

[ 5 0 1 0 3 995 0 0]

[ 14 7 8 9 5 7 1004 13]

[ 15 6 12 12 11 6 12 1005]]

Gewichte in G:

{('s5', 't5'): 995, ('s0', 't5'): 5, ('t6', 's4'): 0, ('s6', 't5'): 7, ('s6', 't3'): 9, ('s3', 't7'): 7, ('t4', 's3'): 0, ('s0', 't6'): 7, ('s6', 't7'): 13, ('s1', 't3'): 2, ('s5', 't0'): 5, ('s3', 't1'): 3, ('s7', 't6'): 12, ('s1', 't2'): 3, ('s0', 't3'): 3, ('s2', 't1'): 2, ('s2', 't7'): 5, ('t1', 's0'): 0, ('s4', 't2'): 4, ('s4', 't5'): 5, ('t5', 's2'): 0, ('s2', 't3'): 2, ('s5', 't1'): 0, ('s4', 't0'): 6, ('s3', 't2'): 4, ('s1', 't1'): 993, ('s6', 't4'): 5, ('s1', 't6'): 0, ('s5', 't7'): 0, ('s7', 't3'): 12, ('s6', 't2'): 8, ('s5', 't6'): 0, ('s5', 't4'): 3, ('s6', 't6'): 1004, ('s3', 't5'): 1, ('s2', 't4'): 1, ('s3', 't3'): 996, ('s4', 't4'): 997, ('s7', 't2'): 12, ('s5', 't2'): 1, ('t7', 's1'): 0, ('s0', 't0'): 1001, ('s6', 't1'): 7, ('s1', 't0'): 0, ('s7', 't4'): 11, ('s7', 't7'): 1005, ('s7', 't5'): 6, ('s6', 't0'): 14, ('s2', 't0'): 1, ('s0', 't2'): 2, ('s3', 't6'): 3, ('s7', 't1'): 6, ('t3', 's5'): 0, ('s4', 't3'): 1, ('s2', 't6'): 0, ('s3', 't0'): 4, ('s2', 't2'): 994, ('s0', 't7'): 9, ('s0', 't4'): 4, ('s4', 't7'): 7, ('s4', 't1'): 3, ('s7', 't0'): 15, ('s1', 't4'): 1, ('s1', 't5'): 0}

J: [2]

vstrich: [7 7 8 7 5 6 6 7]

ustrich: [ 7. 7. 6. 5. 6. 7. 0. inf]

k,P,vk: 2, ['s6', 't2'], 8

u: [41 37 36 37 39 38 39 47]

v: [40 44 44 43 41 42 41 41]

x:

[[0 1 0 0 0 0 0 0]

[0 0 0 0 0 0 0 1]

[0 0 0 0 0 1 0 0]

[0 0 0 0 1 0 0 0]

[0 0 0 0 0 0 1 0]

[0 0 0 1 0 0 0 0]

[0 0 1 0 0 0 0 0]

[0 0 0 0 0 0 0 0]]

P: ['s6', 't2']

######################## Letzte Iteration ########################

ER: [(0, 1), (1, 7), (2, 5), (3, 4), (4, 6), (5, 3), (6, 2)]

EB: [(0, 0), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (2, 6), (2, 7), (3, 0), (3, 1), (3, 2), (3, 3), (3, 5), (3, 6), (3, 7), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 7), (5, 0), (5, 1), (5, 2), (5, 4), (5, 5), (5, 6), (5, 7), (6, 0), (6, 1), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (7, 0), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7)]

c:

[[1001 0 1 3 6 6 8 9]

[ 0 993 2 2 3 1 1 0]

[ 0 1 992 1 2 0 0 4]

[ 2 1 1 994 0 0 2 5]

[ 5 2 2 0 998 5 0 6]

[ 5 0 0 0 5 996 1 0]

[ 7 0 0 2 0 1 998 6]

[ 16 7 12 13 14 8 14 1006]]

Gewichte in G:

{('s5', 't5'): 996, ('s0', 't5'): 6, ('t6', 's4'): 0, ('s6', 't5'): 1, ('s6', 't3'): 2, ('s3', 't7'): 5, ('t4', 's3'): 0, ('s0', 't6'): 8, ('s6', 't7'): 6, ('s1', 't3'): 2, ('s5', 't0'): 5, ('s3', 't1'): 1, ('s7', 't6'): 14, ('s1', 't2'): 2, ('s0', 't3'): 3, ('s2', 't1'): 1, ('s2', 't7'): 4, ('t1', 's0'): 0, ('s4', 't2'): 2, ('s4', 't5'): 5, ('s3', 't5'): 0, ('t5', 's2'): 0, ('s2', 't3'): 1, ('s5', 't1'): 0, ('s4', 't0'): 5, ('s3', 't2'): 1, ('s1', 't1'): 993, ('s6', 't4'): 0, ('s1', 't6'): 1, ('s5', 't7'): 0, ('s7', 't3'): 13, ('s5', 't6'): 1, ('s5', 't4'): 5, ('s6', 't6'): 998, ('t2', 's6'): 0, ('s2', 't4'): 2, ('s3', 't3'): 994, ('s4', 't4'): 998, ('s7', 't2'): 12, ('s5', 't2'): 0, ('t7', 's1'): 0, ('s0', 't0'): 1001, ('s6', 't1'): 0, ('s1', 't0'): 0, ('s7', 't4'): 14, ('s7', 't7'): 1006, ('s7', 't5'): 8, ('s6', 't0'): 7, ('s2', 't0'): 0, ('s0', 't2'): 1, ('s3', 't6'): 2, ('s7', 't1'): 7, ('t3', 's5'): 0, ('s4', 't3'): 0, ('s2', 't6'): 0, ('s3', 't0'): 2, ('s2', 't2'): 992, ('s0', 't7'): 9, ('s0', 't4'): 6, ('s4', 't7'): 6, ('s4', 't1'): 2, ('s7', 't0'): 16, ('s1', 't4'): 3, ('s1', 't5'): 1}

J: [0]

vstrich: [8 7 8 8 8 8 8 8]

ustrich: [7. 8. 8. 8. 8. 8. 8. 0.]

k, P, vk: 0, ['s7', 't5', 's2', 't0'], 8

u: [48 45 44 45 47 46 47 47]

v: [48 51 52 51 49 50 49 49]

x:

[[0 1 0 0 0 0 0 0]

[0 0 0 0 0 0 0 1]

[1 0 0 0 0 0 0 0]

[0 0 0 0 1 0 0 0]

[0 0 0 0 0 0 1 0]

[0 0 0 1 0 0 0 0]

[0 0 1 0 0 0 0 0]

[0 0 0 0 0 1 0 0]]

Die optimale Zuordnung π ist [(0, 1), (1, 7), (2, 0), (3, 4), (4, 6), (5, 3), (6, 2), (7, 5)] mit einer Weglänge w(π) = 30.

**Aufgabe 2 Das Rundreiseproblem**

Um aus dem Zuordnungsproblem eine optimale Lösung des zugehörigen Rundreiseproblems zu generieren, benutzen wir die gegebene Matlab Funktion aus Moodle.

Die Ergebnisse sehen wie folgt aus:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| π (i) | 3 | 1 | 7 | 6 | 4 | 8 | 5 | 2 |
| Kantenlänge | 4 | 3 | 5 | 5 | 4 | 3 | 2 | 4 |

mit w(π) = 30

**Aufgabe 3 Die Nearest-Neighbor-Heuristik**

Das Rundreiseproblem kann man mit dem Nearest-Neighbor Algorithmus ebenfalls lösen.

Die Ergebnisse des Algorithmus lauten:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Knoten Von | 1 | 2 | 8 | 6 | 7 | 5 | 4 | 3 |
| Knoten Zu | 2 | 8 | 6 | 7 | 5 | 4 | 3 | 1 |
| Kantenlänge | 3 | 4 | 3 | 4 | 2 | 4 | 8 | 4 |

mit w(π) = 32