Příloha A. Kód algoritmu v jazyce Python pro ověření 2 skladů v Jažlovicích a Trnavě

```
001: # -*- coding: utf-8 -*-
002:
003:
004: names = ['Poznan', 'Lodz', 'Grodjizk', 'Wroclaw', 'Rzeszow', 'Czeladz', 'Jazlovice', 'Brno', 'Zilina',
005: 'Kosice', 'Zvolen', 'Trnava', 'Budapest']
006: weights = [22889, 6507, 5958, 14486, 43207, 42050, 453746, 37713, 12440, 43084, 5287, 139073, 86041]
007: dist = [[0, 217.91, 285.64, 173.50, 643.05, 401.45, 458.02, 581.48, 491.23, 728.26, 596.34, 649.09, 900.05],
008:         [217.91, 0, 114.36, 222.24, 432.39, 190.79, 558.65, 459.89, 362.46, 517.61, 467.58, 527.50, 778.92],
009:         [285.64, 114.36, 0, 329.22, 289.22, 252.53, 665.63, 521.63, 424.20, 579.34, 529.32, 589.24, 840.66],
010:         [173.50, 222.24, 329.22, 0, 437.02, 205.95, 338.19, 401.90, 311.65, 522.24, 416.77, 469.51, 720.93],
011:         [643.05, 432.39, 289.22, 437.02, 0, 251.03, 683.08, 497.62, 324.59, 200.03, 388.10, 475.17, 459.72],
012:         [401.45, 190.79, 252.53, 205.95, 251.03, 0, 456.41, 270.95, 175.27, 341.86, 280.39, 338.56, 589.98],
013:         [458.02, 558.65, 665.63, 338.19, 683.08, 456.41, 0, 188.37, 395.21, 645.64, 507.03, 368.40, 507.59],
014:         [581.48, 459.89, 521.63, 401.90, 497.62, 270.95, 188.37, 0, 211.36, 461.80, 327.08, 188.44, 327.64],
015:         [491.23, 362.46, 424.20, 311.65, 324.59, 175.27, 395.21, 211.36, 0, 252.15, 107.94, 155.79, 332.11],
016:         [728.26, 517.61, 579.34, 522.24, 200.03, 341.86, 645.64, 461.80, 252.15, 0, 211.79, 361.32, 260.96],
017:         [596.34, 467.58, 529.32, 416.77, 388.10, 280.39, 507.03, 327.08, 107.94, 211.79, 0, 150.69, 157.86],
018:         [649.09, 527.50, 589.24, 469.51, 475.17, 338.56, 368.40, 188.44, 155.79, 361.32, 150.69, 0, 186.92],
019:         [900.05, 778.92, 840.66, 720.93, 459.72, 589.98, 507.59, 327.64, 332.11, 260.96, 157.86, 186.92, 0]
020:         ]
021: france = [828.93, 947.83, 1054.81, 727.37, 1146.57, 919.9, 462.84, 647.02, 850.56, 1102.51, 924.78, 784.76, 897.52]
022:
023: # Krok 1: zvolime vychozi mnozinu dep
024: k = int(input('Zadejte pocet dep: '))
025: D2 = []
026: for i in range(k):
027:     D2.append(int(input('Zadejte depo %s: ' %(i+1))))
028: overall = len(weights)
029: # Urcime mnozinu neprozkoumanych vrcholy
030: N = []
031: for i in range(1, overall+1):
032:     if i not in D2:
033:         N.append(i)
034:
035: #urcime f(D)
036: def fD2(D2):
037:     #zaradime vrcholy do atrakcnich obvodu
038:     A = []
039:     print('Kandidaty pro umisteni depa jsou ', D2)
040:     for i in range(k):
041:         A.append([])
042:     for i in range(overall):
043:         mini = dist[D2[0]-1][i]
044:         which = 0
045:         for c in range(1,k):
046:             if mini >= dist[D2[c]-1][i]:
047:                 mini = dist[D2[c]-1][i]
048:                 which = c
049:         A[which].append(i)
050:     for i in range(k):
051:         norm_A = [x+1 for x in A[i]]
052:         print('Atrakcni obvod pro vrchol %s je : %s' %(D2[i], norm_A))
053:     #vahy v kazdem atrakcnim obvodu A
054:     weightA = []
055:     for i in range(k):
056:         subw = 0
057:         for s in range(len(A[i])):
058:             subw += weights[A[i][s]]
059:         weightA.append(subw)
060:     #vypocteme dopravni praci v kazdem obvodu
061:     result = []
062:     resulto = 0
063:     for c in range(k):
064:         resulto = 0
065:         for i in range(len(A[c])):
066:             resulto += dist[D2[c]-1][A[c][i]] * weights[A[c][i]]
067:         resulto += france[D2[c]-1]*weightA[c]
068:         result.append(resulto)
069:     print('dopravni prace pro vrcholy %s je: %s' %(D2, sum(result)))
070:     return sum(result)
071:
072:
073: #vysledky
074: def printres(D2):
075:     print("Mnozina D { %s }: celkova dopravni prace cini %s" %(D2, fD2(D2)))
076:
077:
078: minimum = fD2(D2)
079:
```

```
080: #tělo algoritmu
081:
082: #krok 2
083: while True:
084:     z = 0
085:     #Je-li N = prazdna mnozina?
086:     while N != []:
087:         #pokud ne, vybereme libovolny bod a vypocteme f(D)
088:         print('Zvolime vrchol %s z mnoziny neprozkoumanych vrcholy N' %(N[0]))
089:         hubNew = N[0]
090:         res = []
091:         D = []
092:         for i in range(k):
093:             D.append([])
094:         for c in range(k):
095:             D[c] = D2[:]
096:             D[c][c] = hubNew
097:             res.append(fD2(D[c]))
098:         #krok 3
099:         locmin = res[0]
100:         which = 0
101:         for c in range(1,k):
102:             if locmin >= res[c]:
103:                 locmin = res[c]
104:                 which = c
105:         print('Urcime minimalni dopravni praci mezi dvojici vrcholu %s. Minimum je %s ' %(D, locmin))
106:         #porovname hodnoty
107:         if locmin >= minimum:
108:             N.remove(hubNew)
109:             print('Nove minimum je vetsi nez predchozi, proto vrchol %s vyloucime z mnoziny N ' %hubNew)
110:         else:
111:             D2 = D[which][:]
112:             z += 1
113:             minimum = locmin
114:             print('Nove minimum je mensi nez predchozi, proto vytvorime novou mnozinu dep Dk ', D2)
115:             break
116:     #krok 4
117:     if z == 0:
118:         break
119:     else:
120:         N = []
121:         for i in range(1, overall+1):
122:             if i not in D2:
123:                 N.append(i)
124: #krok 5
125: resName = []
126: for i in range(k):
127:     resName.append(names[D2[i]-1])
128: print("Mnozina D { %s } predstavuje vrcholove optimalni rozmisteni %s dep na siti. "
129:       "Minimalni hodnota dopravni prace je %s . Vrcholy: %s" %(resName, k, minimum, D2))
```