**Input**: V1, V2, last, first

**Output**: path and min\_lt

// block 1. subtracting lists

V = V1 - V2

N = len(V)

**while** N>0 **do**

// 2. Clearing V from fragments of the path that do not start from the initial vertex

**for** iϵ{1..N} **do**

**if** Vi.first\_vertix!=first **then**

delete (Vi);

**end**

**end**

//3. Search for new path fragments

// Passage through list V

**for** iϵ{1.. length(V)} **do**

// Passage through list V1

**for** jϵ{1.. length(V1)} **do**

**if** Vi = V1j **then**

// Searching for the continuation of the edge (V1i, V2i)

// Passage through list V1

**for** lϵ{1.. length(V1)} **do**

**if** V2j == V1l **then**

// adding an edge (V1l, V2l) to the Path (first, V2j)

V1\_add.append(V1l)

V2\_add.append(V2l)

# updating the path length

lt\_add.append(ltj+ltl)

**end**

**end**

**end**

**end**

**end**

//4. Removing the paths that end at the last vertex from the lists

// Passage through list V1\_add

**for** iϵ{1.. length(V2\_add)} **do**

**if** V2i == last **then**

final\_path.add (V1\_addi+V2\_addi); final\_lt.add(lt\_addi)

delete (V2\_addi);

delete (V1\_addi);

delete (lt\_addi)

**end**

**end**

// 5. Looking for a minimum length path

**for** iϵ{1.. length(final\_lt)} **do**

// find a minimum length path and memorize it: path and min\_lt

**End**

// 6. Update V1 and V2 lists

V1=V1\_add+V1

V2=V2\_add+V2

lt=lt\_add+lt

**end**