

Těžký příklad — Babička v MHD

Vaše babička by ráda cestovala po Praze a jejím okolí za svými vnoučaty nebo za kámoškami. Nicméně to má jeden zásadní problém, bolí ji nohy a chůze při přestupech mezi linkami je pro ni velmi náročná. Naopak sezením v tramvaji či metru se neunaví vůbec. Babička je v důchodu, takže má spoustu času — je jí úplně jedno, jestli cesta trvá 20 minut nebo hodinu, pokud se vyhne zbytečnému přestupování.

Vaším úkolem je naprogramovat vyhledávač spojení, který najde trasu s **minimálním počtem přestupů**. Proto v této úloze musíte pracovat se **stavovým prostorem**, kde cena přechodu není vzdálenost ani čas, ale změna dopravního prostředku.

Napište program **pid.py**, který načte data o síti PID z XML souboru, přečte konkrétní cestovní požadavek z textového souboru a najde trasu s minimálním počtem přestupů mezi zadanými stanicemi, přičemž bude respektovat omezení na povolené dopravní prostředky a denní dobu.

- **Vstup:**

- Program se spouští z příkazové řádky ve formátu:

```
python3 pid.py <data.xml> <ukol.txt>
```

- **1. argument:** Název XML souboru s definicí sítě (např. `pid.xml`). Tento soubor obsahuje seznam stanic, jejich polohu a linky, které na nich staví.
- **2. argument:** Název textového souboru s konkrétním zadáním cesty (např. `task_01.txt`). Tento soubor obsahuje čtyři řádky, kde každý řádek má následující význam:
 1. Jméno startovní stanice (např. `Dejvická`)
 2. Jméno cílové stanice (např. `I. P. Pavlova`)
 3. Denní doba (`den` nebo `noc`)
 4. Seznam povolených dopravních prostředků oddělených čárkou (např. `bus, tram, metro`)

- **Výstup:**

- Pokud cesta existuje, vypište počet přestupů jako jedno číslo na standardní výstup.
- Pokud cesta nexistuje, vypište `-1`

Poznámky k implementaci:

- **Stavový prostor:** Uvědomte si, že "být na zastávce Muzeum" není úplná informace. Babička je buď "na zastávce Muzeum v Metru A" nebo "na zastávce Muzeum v Metru C". Přechod mezi těmito stavy (přestup) ji stojí úsilí. Jízda z "Muzeum (A)" na "Můstek (A)" ji nestojí nic.
- **XML Parsing:** Pro načtení dat použijte knihovnu `xml.etree.ElementTree`. Soubor obsahuje pouze stanice, nikoliv pořadí zastávek na lince. Předpokládejte, že pokud dvě stanice sdílí stejné ID linky (např. obě mají linku "A"), lze mezi nimi přejet. Data můžete načíst například takto:

```

import xml.etree.ElementTree as ET

with open(filename, 'r', encoding='utf-8') as file:
    tree = ET.parse(file)
    root = tree.getroot()

    for group in root.findall('group'):
        #group.attrib je dictionary, kde klíče jsou jména atributu
        jmenoZastavky = group.attrib["name"] #nacteme atribut name z tagu group

        for stop in group.findall('stop'):
            #stop.attrib je dictionary, kde klíče jsou jména atributu
            if stop.attrib["mainTrafficType"] == "undefined":
                continue #ignorujeme podivne nastupiste

            for line in stop.findall('line'):
                #nacteme jméno spoje z atributu name z tagu line
                jmenoSpoje = line.attrib["name"]
                typSpoje = line.attrib["type"]
                isNight = False
                #line.attrib je dictionary, kde klíče jsou jména atributu
                if "isNight" in line.attrib and line.attrib["isNight"] == 'true':
                    isNight = True
                print("Zastavkou", jmenoZastavky, "projizdi spoj", jmenoSpoje,
                    "typu", typSpoje, "nocni=",isNight)

```

- Zastávky (stanice) jsou reprezentovány jako elementy <group ... > s atributem name.
- Některé názvy zastávek <group name=...> se mohou vyskytovat v XML vícekrát, v takovém případě se jedná o stejnou zastávku a platí pro ni všechny linky, které jsou pro příslušné zastávky uvedeny.
- Nástupiště <stop ... > s atributem mainTrafficType="undefined" ignorujeme.

• Filtrování:

- Při hledání cesty zohledněte denní dobu (3. řádek zadání). Například pokud je zadáno "noc", musíte ignorovat linky metra, autobusy a tramvaje, které v noci nejezdí. Linky, které jezdí pouze v noci mají v elementu line atribut isNight nastaven na hodnotu 'true'. Pro kontrolu jak je nastaven atribut isNight můžete použít následující kód:

```

if "isNight" in line.attrib and line.attrib["isNight"] == 'true':
    print("Noční linka")

```

- Při načítání dat ignorujte dopravní prostředky, které nejsou uvedeny ve 4. řádku zadání. Tím simulujeme situaci, kdy jsou některé linky mimo provoz například z důvodu výluky. Možné dopravní prostředky jsou: bus, tram, metro, train, ferry, trolleybus.

Testovací data

Pro testování vašeho programu můžete použít příložený XML soubor `pid.xml`, který jsme získali z otevřených dat Pražské integrované dopravy (PID) <https://pid.cz/en/opendata/>. Nicméně je potřeba zmínit, že tento soubor byl zjednodušen a upraven pro potřeby této úlohy. Pro testování na Brute je soubor téměř identický s originálem z PID, pouze jsme upravili název jedné zastávky (Jirny,LE&CO na Jirny,LEaCO), která obsahovala nepovolený znak `&`.

Co je xml

XML je textový formát pro zápis strukturovaných dat. Hlavní elementy jsou tzv. tagy, které jsou značeny `<tag ... />` nebo `<tag> ... </tag>`. Součástí tagu mohou být atributy:

```
<tag atribut1="hodnota1" atribut2="hodnota2"/>
```

Tagy lze seskupovat uvnitř jiných tagů, například:

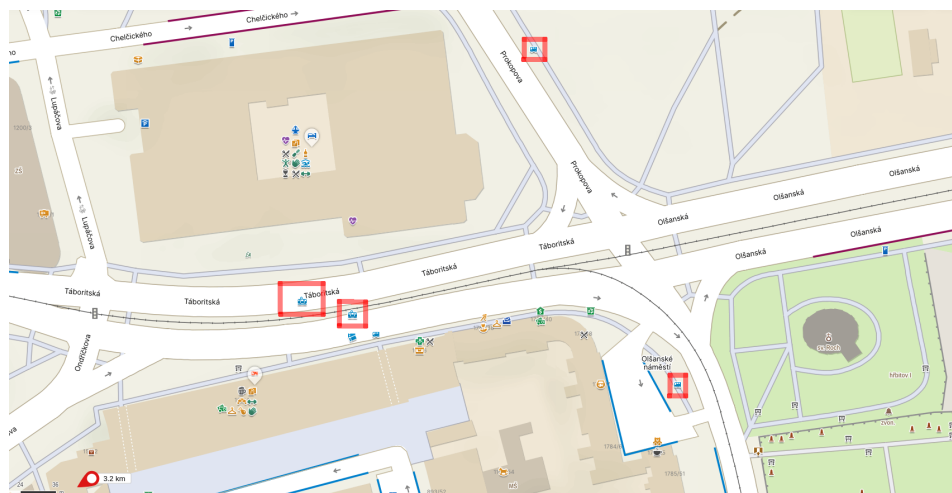
```
<group name="jmeno zastavky" dalsiAtribut="nejakaHodnota">
  <stop name="oznaceni nastupiste" dalsiAtribut="dalsi hodnota">
    <line name="jmeno spoje1" type="typ spoje" />
    <line name="jmeno spoje2" type="typ spoje" />
  </stop>
  <stop name="oznaceni nastupiste" dalsiAtribut="dalsi hodnota">
    <line name="jmeno spoje3" type="typ spoje" />
    <line name="jmeno spoje4" type="typ spoje" />
  </stop>
</group>
```

Čtení XML souborů je netriviální a proto používáme knihovnu `xml.etree.ElementTree`. V našem úkolu ji použijeme pro načtení celého XML a následně projdeme všechny zastávky a pro každou zastávku všechny nástupiště a pro každé nástupiště všechny spoje.

1 Vstupní data

Pojem “zastávka” (nebo též “stanice”) může ve městech znamenat jak jedno konkrétní místo pro nástup do dopravního prostředku, tak i několik takových míst — nástupišť (například nástupiště pro různé dopravní prostředky, noční a denní provoz, spoje náhradní dopravy, ...). Rozlišujeme pojem “stanice/zastávka” a “nástupiště”. Zatímco nástupiště označuje konkrétní místo pro nástup nebo výstup z dopravního prostředku, pojem zastávka označuje jednu nebo více nástupišť.

Například na Olšanském náměstí (zastávka “Olšanské náměstí”) v síti PID je několik nástupišť, které jsou červeně vyznačené na obrázku: dvě pro tramvaje a dvě pro autobusy.



Zastávka se v xml souboru označuje tagem `<group name="jmeno zastavky" .. />`, kde atribut `name` udává její jméno. Každé nástupiště je v xml souboru uvedeno v tagu `<stop ... />`, například `"<stop id="identifikator" ... >"`, a pokud tímto konkrétním místem/nástupištěm projíždí nějaký spoj, je tento spoj uveden v tagu `<line name="jmenoLinky" .. >`.

V našem příkladu jsou pro cestující určeny místa s interním identifikátorem 516/1 a 516/2 (pro tramvaje) a dále dvě místa 516/3 a 516/4 pro autobusy. Dále je v souboru definována místo (516/51), které ale není aktuálně využíváno (takové místo může být využíváno například v případě výluky pro zastavení spoje náhradní dopravy), v souboru je toto místo označeno `mainTrafficType="undefined"`. Poslední místo je 516/52, které je určeno pro noční autobus 898.

Pokud chceme zjistit, které spoje projíždějí zastávkou (stanicí) Olšanské náměstí, je třeba v xml souboru vyhledat tag `group`, jehož atribut `name="Olšanské náměstí"`, a potom vyhledat všechny přidružené nástupiště (tag `<stop ../>`) a v každém z nich dále všechny linky (tag `<line .. />`).

```
<group name="Olšanské náměstí" districtCode="AB" idosCategory="301003" idosName="Olšanské náměstí"
  fullName="Olšanské náměstí" uniqueName="Olšanské náměstí" node="516" cis="59186" avgLat="50.08257"
  avgLon="14.45814" avgJtskX="-740268.6" avgJtskY="-1043911.5" municipality="Praha" mainTrafficType="tram">
<stop id="516/1" platform="A" altIdosName="Olšanské náměstí" lat="50.08249" lon="14.4574671"
jtskX="-740317.563" jtskY="-1043913.75" zone="P" mainTrafficType="tram" wheelchairAccess="possible" gtfsIds="U516Z1P">
  <line id="5" name="5" type="tram" direction="Slivenec"/>
  <line id="7" name="7" type="tram" direction="Bílá labuť"/>
  <line id="9" name="9" type="tram" direction="Sídliště Řepy"/>
  <line id="15" name="15" type="tram" direction="Kotlářka"/>
  <line id="26" name="26" type="tram" direction="Dědina"/>
  <line id="95" name="95" type="tram" isNight="true" direction="Ústřední dílny DP" direction2="Bohemians"/>
  <line id="98" name="98" type="tram" isNight="true" direction="Sídliště Řepy"/>
</stop>
<stop id="516/2" platform="B" altIdosName="Olšanské náměstí" lat="50.08245" lon="14.4577379"
jtskX="-740299" jtskY="-1043920.88" zone="P" mainTrafficType="tram" wheelchairAccess="possible" gtfsIds="U516Z2P">
  <line id="5" name="5" type="tram" direction="Lehovec"/>
  <line id="7" name="7" type="tram" direction="Radlická"/>
  <line id="9" name="9" type="tram" direction="Spojovací"/>
  <line id="15" name="15" type="tram" direction="Sídliště Ďáblice"/>
  <line id="26" name="26" type="tram" direction="Nádraží Hostivař"/>
  <line id="95" name="95" type="tram" isNight="true" direction="Vozovna Kobylisy"/>
  <line id="98" name="98" type="tram" isNight="true" direction="Spojovací"/>
</stop>
<stop id="516/3" platform="C" altIdosName="Olšanské náměstí" lat="50.0823021" lon="14.4592342"
jtskX="-740195.2" jtskY="-1043951.75" zone="P" mainTrafficType="bus" wheelchairAccess="unknown" gtfsIds="U516Z3P">
  <line id="136" name="136" type="bus" direction="Jižní Město"/>
  <line id="175" name="175" type="bus" direction="Háje"/>
  <line id="913" name="913" type="bus" isNight="true" direction="Točná"/>
  <line id="914" name="914" type="bus" isNight="true" direction="Smíchovské nádraží" direction2="Budějovická"/>
</stop>
<stop id="516/4" platform="D" altIdosName="Olšanské náměstí" lat="50.08314" lon="14.4586782"
jtskX="-740221.9" jtskY="-1043853.63" zone="P" mainTrafficType="bus" wheelchairAccess="unknown" gtfsIds="U516Z4P">
  <line id="136" name="136" type="bus" direction="Sídliště Čakovice"/>
  <line id="175" name="175" type="bus" direction="Florenc"/>
  <line id="913" name="913" type="bus" isNight="true" direction="Dolní Chabry"/>
  <line id="914" name="914" type="bus" isNight="true" direction="Vinoř"/>
</stop>
<stop id="516/51" platform="Z1" altIdosName="Olšanské náměstí" lat="50.08264" lon="14.4578943"
jtskX="-740285" jtskY="-1043901" zone="P" mainTrafficType="undefined" wheelchairAccess="unknown" gtfsIds="U516Z51P"/>
<stop id="516/52" platform="Z2" altIdosName="Olšanské náměstí" lat="50.0823936" lon="14.45783"
jtskX="-740293.25" jtskY="-1043928.06" zone="P" mainTrafficType="bus" wheelchairAccess="unknown" gtfsIds="U516Z52P">
  <line id="898" name="X98" type="bus" isNight="true" direction="Nákladové nádraží Žižkov"/>
</stop>
</group>
```

Na první pohled je XML soubor komplikovaný, ale poskytuje řadu užitečných vlastností, a proto se často používá pro výměnu dat. Jednou z důležitých vlastností je možnost strukturovat data (tzn. tagy mohou být vnořeny), dále je možné informace vkládat do atributů a při načítání použít jen ty atributy, které aktuálně potřebujeme. Například ve výše uvedeném příkladu si všimněte, že v definici zastávky je řada dalších užitečných informací (které ale v naší domácí úloze nepotřebujeme): například poloha a výška zastávky, nebo její identifikace v systému IDOS (atributy `altIdosName` a `idosCategory`). Jednotlivá nástupiště také mají detailnější informace, například polohu nebo informaci o tom, jestli jsou vhodné pro vozíčkáře.

1.1 Part 1: Jeden dopravní prostředek ve dne

Vstupní soubor obsahuje:

```
Zborovská  
Flora  
den  
tram
```

Vstup

```
python3 pid.py pid.xml task_0.txt
```

Výstup programu

```
0
```

Komentář: Možné cesty jsou:

Start na 'Zborovská' (linka 16) - Cíl: 'Flora';

Start na 'Zborovská' (linka 7) - Cíl: 'Flora'.

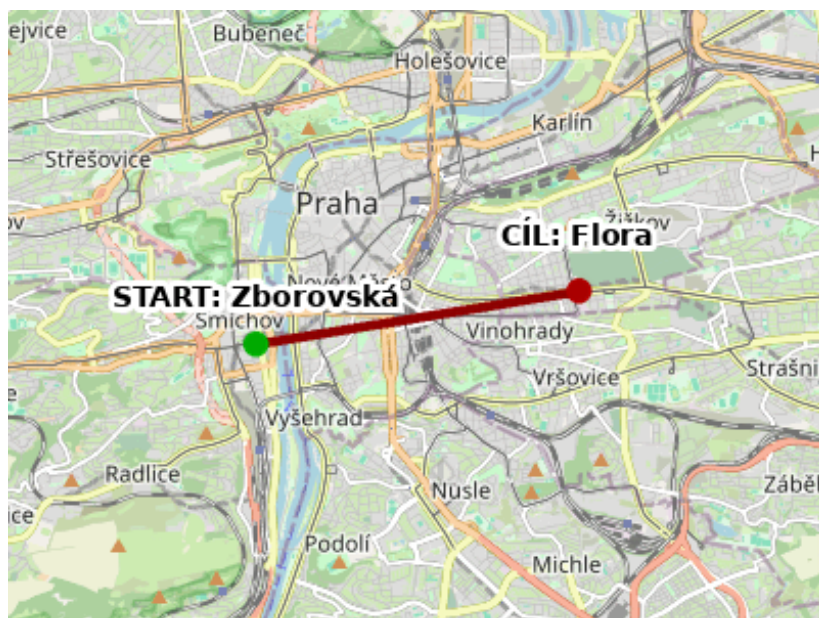


Figure 1: Mapa trasy

1.2 Part 2: Jeden dopravní prostředek v noci

Vstupní soubor obsahuje:

```
Dejvická  
Letenské náměstí  
noc  
tram
```

Vstup

```
python3 pid.py pid.xml task_20.txt
```

Výstup programu

```
0
```

Komentář: Možné cesty jsou:

Start na 'Dejvická' (linka 91) - Cíl: 'Letenské náměstí'.



Figure 2: Mapa trasy

1.3 Part 3: Metro bez přestupu

Vstupní soubor obsahuje:

```
Střížkov  
Muzeum  
den  
metro
```

Vstup

```
python3 pid.py pid.xml task_40.txt
```

Výstup programu

```
0
```

Komentář: Možné cesty jsou:

Start na 'Střížkov' (linka C) - Cíl: 'Muzeum'.

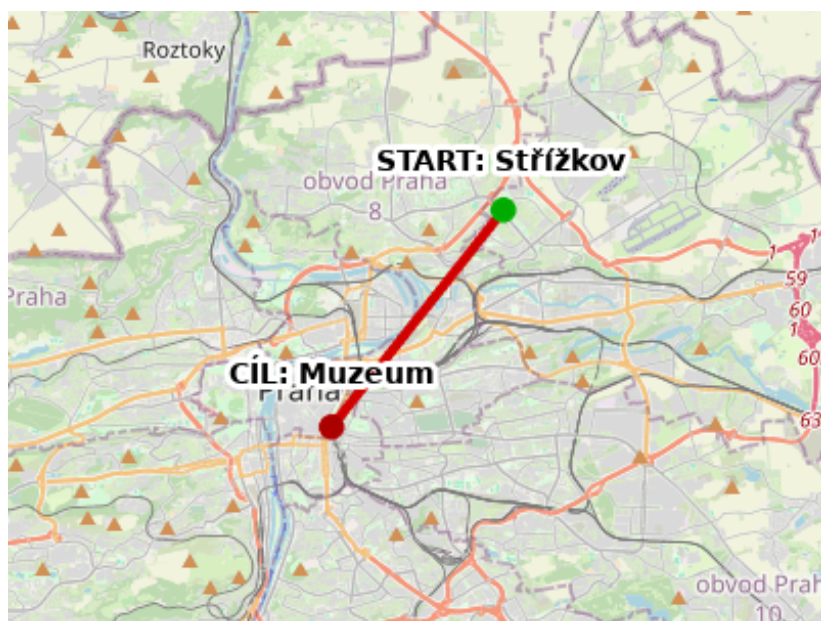


Figure 3: Mapa trasy

1.4 Part 4: Tramvaj bez přestupu

Vstupní soubor obsahuje:

```
Okrouhlická  
Moráň  
den  
tram
```

Vstup

```
python3 pid.py pid.xml task_60.txt
```

Výstup programu

```
0
```

Komentář: Možné cesty jsou:

Start na 'Okrouhlická' (linka 24) - Cíl: 'Moráň';

Start na 'Okrouhlická' (linka 3) - Cíl: 'Moráň'.

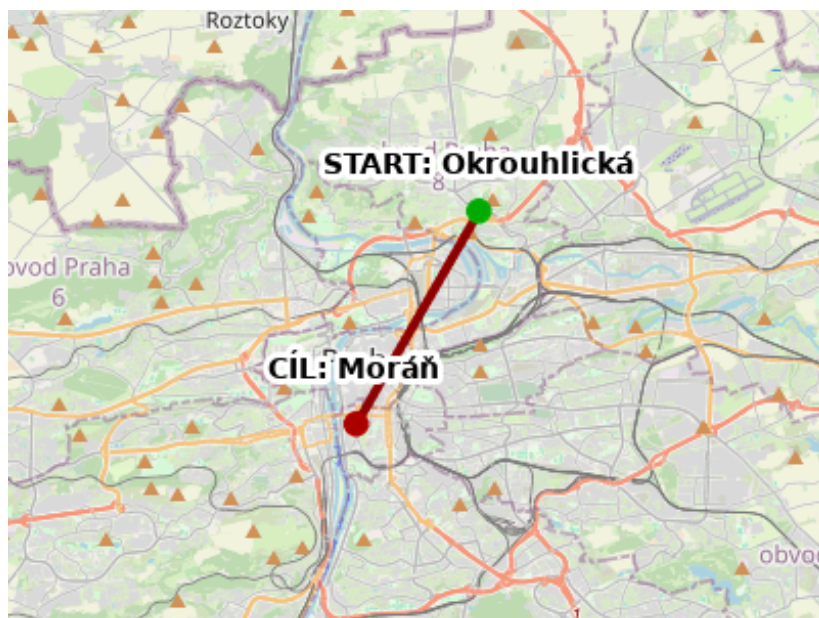


Figure 4: Mapa trasy

1.5 Part 5: Tramvaj s přestupem

Vstupní soubor obsahuje:

```
Otakarova  
Sídliště Libuš  
den  
tram
```

Vstup

```
python3 pid.py pid.xml task_80.txt
```

Výstup programu

```
1
```

Komentář: Možné cesty jsou:

Start na 'Otakarova' (linka 24) - přestup na 'Březiněveská' (linka 17) - Cíl: 'Sídliště Libuš'.



Figure 5: Mapa trasy

1.6 Part 6: Autobus s přestupem

Vstupní soubor obsahuje:

```
Hradební  
Poliklinika Zahradní Město  
den  
bus
```

Vstup

```
python3 pid.py pid.xml task_120.txt
```

Výstup programu

```
2
```

Komentář: Možné cesty jsou:

Start na 'Hradební' (linka 194) - přestup na 'Florenc' (linka 133) - přestup na 'Sídliště Malešice' (linka 195)
- Cíl: 'Poliklinika Zahradní Město'.

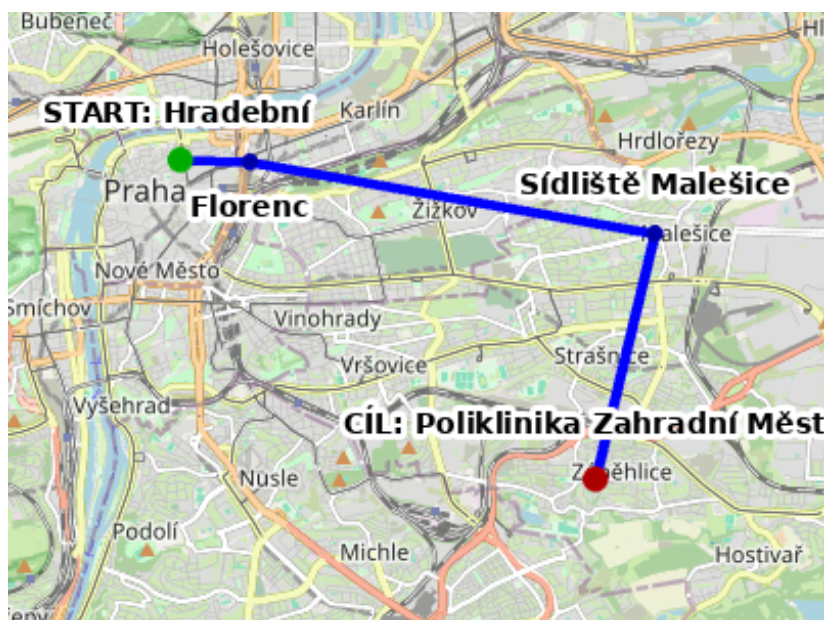


Figure 6: Mapa trasy

1.7 Part 7: Náročný přesun v noci s přestupem

Vstupní soubor obsahuje:

```
Brandýs n.L.-St.Bol.,U Brány  
Křeslice  
noc  
tram,trolleybus,bus
```

Vstup

```
python3 pid.py pid.xml task_160.txt
```

Výstup programu

```
3
```

Komentář: Možné cesty jsou:

Start na 'Brandýs n.L.-St.Bol.,U Brány' (linka 953) - přestup na 'Vinořský hřbitov' (linka 914) - přestup na 'Brumlovka' (linka 905) - přestup na 'Mokrá' (linka 906) - Cíl: 'Křeslice'.



Figure 7: Mapa trasy

1.8 Part 8: Všechny dopravní prostředky s přestupem

Vstupní soubor obsahuje:

```
Litovický potok  
Chyňava,sídliště  
den  
train,tram,metro,trolleybus,bus,ferry
```

Vstup

```
python3 pid.py pid.xml task_200.txt
```

Výstup programu

```
2
```

Komentář: Možné cesty jsou:

Start na 'Litovický potok' (linka 108) - přestup na 'Divoká Šárka' (linka 399) - přestup na 'Kladno,gymnasium' (linka 630) - Cíl: 'Chyňava,sídliště';

Start na 'Litovický potok' (linka 168) - přestup na 'Kukulova' (linka 380) - přestup na 'Loděnice' (linka 636) - Cíl: 'Chyňava,sídliště'.

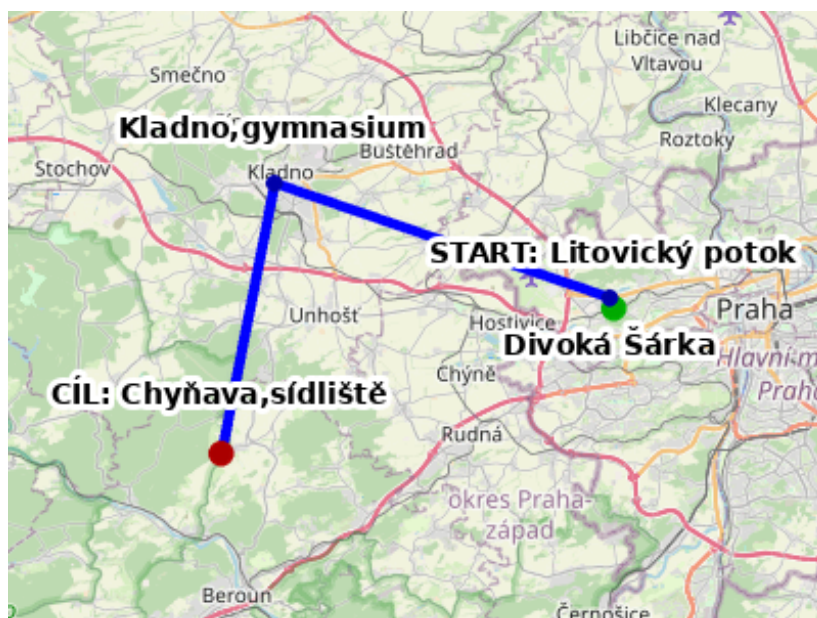


Figure 8: Mapa trasy

1.9 Part 9: Všechny dopravní prostředky s velkým množstvím přestupů

Vstupní soubor obsahuje:

```
Dunajecská  
Chaloupky  
den  
train, tram, metro, trolleybus, bus, ferry
```

Vstup

```
python3 pid.py pid.xml task_280.txt
```

Výstup programu

```
4
```

Komentář: Možné cesty jsou:

Start na 'Dunajecská' (linka 102) - přestup na 'Šimůnkova' (linka 183) - přestup na 'Háje' (linka 517) - přestup na 'Příbram, Jiráskovy sady' (linka 531) - přestup na 'Hořovice, Palackého nám.' (linka 544) - Cíl: 'Chaloupky'.

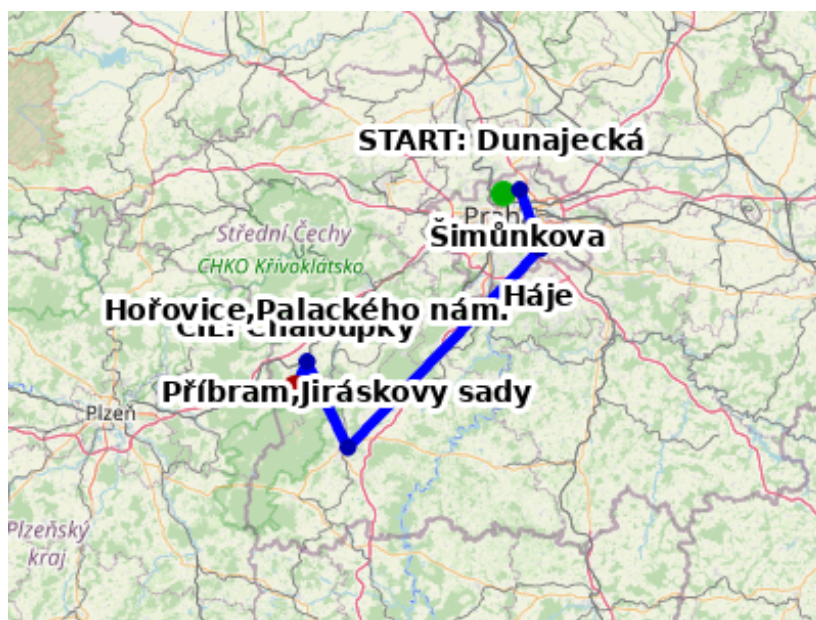


Figure 9: Mapa trasy

1.10 Part 10: Neexistující cesta

Vstupní soubor obsahuje:

```
Kladno,Rakovnická  
Hodonín  
den  
train,tram,metro,trolleybus,bus,ferry
```

Vstup

```
python3 pid.py pid.xml task_380.txt
```

Výstup programu

```
-1
```

1.11 Part 11: Neexistující cesta metrem, trolejbusem nebo přívozem

Vstupní soubor obsahuje:

```
Výstaviště Letňany  
Králov Dvůr,U Tří zvonků  
den  
metro,trolleybus,ferry
```

Vstup

```
python3 pid.py pid.xml task_390.txt
```

Výstup programu

```
-1
```