

## Kontext

---

Jedna z klasických aplikací teorie grafů je v implementaci mapově orientovaných aplikací. Ty jsou, zvláště v dnešní době, velmi žádanými doplňky aplikací hlavně v oblasti mobilních zařízení. Mnohé aplikace doplňují své služby o geografické informace a umožňují uživatelům zobrazovat a vyhledávat dle těchto dat. Z tohoto důvodu si v následujícím projektu vyzkoušíme tvorbu jednoduchého navigačního systému pomocí otevřeného mapového systému OpenStreetMap, který mnohé z mobilních aplikací také využívají.

## Zadání

---

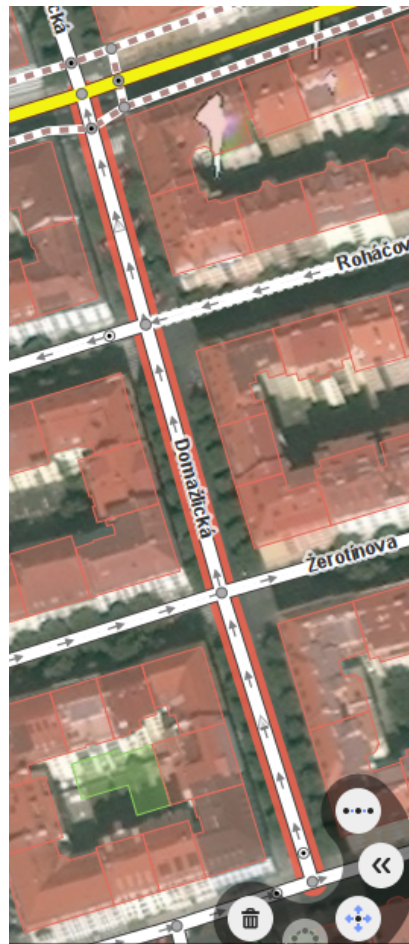
Úkolem tohoto projektu je vytvoření jednoduchého navigačního systému, pro který budeme využívat datových podkladů z OpenStreetMap. Jedná se o crowd-sourcingový mapový systém, jehož obsah vytvářejí přímo jeho uživatelé.

- [www.openstreetmap.org](http://www.openstreetmap.org)

Tento systém poskytuje komplexní mapové podklady, ze kterého využijeme síť ulic. Ulice jsou definovány jednotlivými body, které určují geografickou polohu každé ulice.

Konkrétní mapu by bylo možné zjednodušeně reprezentovat grafem tak, že za vrcholy budou odpovídat křižovatkám (nebo slepým koncům ulic) a hrany budou reprezentovat ulice tyto křižovatky spojující. Tato reprezentace ovšem nezahrnuje cenné informace o topologii sítě – takto by například geografická vzdálenost křižovatek nemusela odpovídat celkové délce ulice z důvodu značně velkého zakřivení jejího běhu. Navíc při této reprezentaci není možné definovat přesněji cílové místo pro náš navigační systém.

Z tohoto důvodu jsou ulice v systému OpenStreetMap reprezentovány posloupností bodů vedoucí od jejího jednoho konce na druhý. Některé body mohou být pouze definicí zakřivení, jiné mohou být křižovatkami s jinými ulicemi, a některé mohou mít ještě další speciální vlastnosti. Příklad takové ulice je zobrazen na následujícím obrázku.



**Obrázek 1:** Příklad části ulice Domažlická poblíž Parukářky, spojující šest bodů, z nichž jsou čtyři zároveň křižovatkami s jinými ulicemi (reprezentované body 21311329, 25973241, 25973242, 25973237) a dva body jsou přechody pro chodce (1131753606 a 1131753673).

Odpovídající struktura v OSM formátu vypadá následovně.

```

1  <way id="39120883" visible="true" version="2" changeset="7160314" timestamp="2011-02-02T00:06:24Z" user=
2  <nd ref="21311329"/>
3  <nd ref="1131753606"/>
4  <nd ref="25973241"/>
5  <nd ref="25973242"/>
6  <nd ref="1131753673"/>
7  <nd ref="25973237"/>
8  <tag k="highway" v="residential"/>
9  <tag k="name" v="Domažlická"/>
10 <tag k="oneway" v="yes"/>
11 </way>
12 <node id="21311329" visible="true" version="2" changeset="766878" timestamp="2009-03-08T23:54:24Z" use
13 <node id="1131753606" visible="true" version="1" changeset="7160314" timestamp="2011-02-02T00:06:12Z"
14 <tag k="highway" v="crossing"/>
15 <tag k="source" v="bing:ortofoto"/>
16 </node>
17 <node id="25973241" visible="true" version="1" changeset="218259" timestamp="2007-02-17T23:29:05Z" use
18 <node id="25973242" visible="true" version="2" changeset="766878" timestamp="2009-03-08T23:54:24Z" use
19 <node id="1131753673" visible="true" version="1" changeset="7160314" timestamp="2011-02-02T00:06:13Z"
20 <tag k="highway" v="crossing"/>
21 <tag k="source" v="bing:ortofoto"/>
22 </node>
23 <node id="25973237" visible="true" version="2" changeset="7160314" timestamp="2011-02-02T00:06:15Z" us

```

V OpenStreetMap se vyskytují tři typy objektů, z nichž objekty typu area pro nás nejsou zajímavé, zbývající dva typy však ano:

- Node <nd>
  - geometrický bod
  - má ID jednoznačné v rámci bodů
  - poloha je definovaná dvojicí Latitude (lat=) a Longitude (lon=)
  - může obsahovat další doplňující informace

- Way <way>
  - seřazený seznam 2 a více objektů typu Node
  - v mapě reprezentuje jakoukoliv lomenou čáru
  - obsahuje další doplňující informace, zejména jakého typu čára je

## Vstupní data

Datový formát používaný pro zaslání map je označován jako OSM a jedná se o XML formát. Základem je tag <osm>, který obsahuje základní informace o licenci a typu generátor

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <osm version="0.6" generator="CGImap 0.4.0 (10746 thorn-05.openstreetmap.org)"
3   copyright="OpenStreetMap and contributors" attribution="http://www.openstreetmap.org/copyright"
4   license="http://opendatacommons.org/licenses/odbl/1-0/">
5 </osm>

```

Pokud byla mapa, kterou zpracováváte, stáhnuta jako podoblast definována obdélníkem, bude datový zdroj obsahovat také následující tag definující tyto hranice pomocí minima a maxima zeměpisných šířek a délek:

```

1 <bounds minlat="50.0864200" minlon="14.4607100" maxlat="50.0888400" maxlon="14.4655600"/>

```

Základní stavební jednotkou, jak je již zmíněno výše, je objekt typu Node. Příklad jsme již viděli:

```

1 <node id="21311329" visible="true" version="2" changeset="766878" timestamp="2009-03-08T23:54:24Z"
2   user="BiIbo" uid="3516" lat="50.0874984" lon="14.4641165"/>

```

Tento tag obsahuje následující informace:

- id ... jednoznačný identifikátor vzhledem k vrcholům
- visible ... nastavení zobrazitelnosti na mapě
- version ... informaci o verzi dat
- changeset ... označení změn - založeno kvůli možnostem editace
- timestamp ... informaci o zanesení poslední informace
- user ... informaci o uživateli, který zanesl danou informaci
- uid ... číselné označení uživatele
- lat ... zeměpisná šířka daného bodu
- lon ... zeměpisná délka daného bodu

Kromě této struktury může bod obsahovat i složitější informace v komplexním tagu. Viděli jsme například přechod pro chodce

```

1 <node id="1131753606" visible="true" version="1" changeset="7160314" timestamp="2011-02-02T00:06:12Z" us
2   <tag k="highway" v="crossing"/>
3   <tag k="source" v="bing:ortofoto"/>
4 </node>

```

nebo jakékoli jiné bodové informace, například

```

1 <node id="290558370" visible="true" version="5" changeset="20938887" timestamp="2014-03-05T21:20:23Z"
2   user="Christoph Lotz" uid="47978" lat="49.8346891" lon="9.8906709">
3   <tag k="addr:city" v="Veitshöchheim"/>
4   <tag k="addr:country" v="DE"/>
5   <tag k="addr:housenumber" v="3"/>
6   <tag k="addr:postcode" v="97209"/>
7   <tag k="addr:street" v="Danziger Straße"/>
8   <tag k="amenity" v="bank"/>
9   <tag k="atm" v="yes"/>
10  <tag k="email" v="VeitshoechheimII@Sparkasse-Mainfranken.de"/>
11  <tag k="fax" v="+49 (0)931 / 382-2780"/>
12  <tag k="name" v="Sparkasse Mainfranken"/>
13  <tag k="phone" v="+49 (0)931 / 382-0"/>
14  <tag k="website" v="https://www.sparkasse-mainfranken.de/module/ihre_sparkasse/filialfinder/index.php
15    ?n=%2Fmodule%2Fihre_sparkasse%2Ffilialfinder%2Fdetails/13093"/>
16  <tag k="wheelchair" v="limited"/>
17 </node>

```

Jak vidíme, tak tyto informace se týkají hlavně firem či zajímavých míst umístěných na tomto místě potažmo adrese.

Dále obsahuje mapa informace o samotných ulicích pomocí tagu <way>, jak už jsme viděli s ulicí Domažlická:

```

1 <way id="39120883" visible="true" version="2" changeset="7160314" timestamp="2011-02-02T00:06:24Z" user
2 <nd ref="21311329"/>
3 <nd ref="1131753606"/>
4 <nd ref="25973241"/>
5 <nd ref="25973242"/>
6 <nd ref="1131753673"/>
7 <nd ref="25973237"/>
8 <tag k="highway" v="residential"/>
9 <tag k="maxspeed" v="50"/>
10 <tag k="name" v="Domažlická"/>
11 <tag k="oneway" v="yes"/>
12 </way>

```

Ulice tak obsahuje jednak základní informace zahrnující

- id ... unikátní ID vzhledem k jednotlivým tagům way
- visible ... informaci o viditelnosti
- changeset ... označení změnové řady
- timestamp ... informaci o datumu změny
- user ... jméno uživatele
- uid ... id uživatele

Poté obsahuje každá ulice informaci o posloupnosti jednotlivých nodů v tagu `<nd>`. Atribut `ref` odkazuje na id specifického Node, který je součástí ulice. Ulice výše se tak skládá ze 6 nodů (dvou počátečních a čtyř, které jsou součástí dané ulice). Následně jsou na ulici různé typy dalších vlastností jako sekvence tagů pojmenovaných `<tag>`. Výše zmíněné jsou

- `highway` -- jedná se o silnici nebo cestu
- `residential` -- ulice v obydlené části
- maximální rychlost
- jméno dané ulice
- informace, že je ulice jednosměrná (pokud není, tento tag chybí)

Je potřeba dát pozor, že ne vše uložené v systému jako way je ve skutečnosti ulicí. Toto označují různé tagy jako `amenity`, `notroad` či `building`. Pro správné načtení sítě ulic prostudujte dokumentaci OpenStreetMap a zkuste si část mapy Prahy stáhnout (pozor, XML soubor větší oblasti je velmi rozsáhlý).

## Úkol

Úkolem tohoto projektu je vytvořit jednoduchou navigaci, která načte data ze systému OpenStreetMap, bude umět vyhledat nejkratší cestu mezi cílovými body a výsledek zobrazit v systému **Graphviz**. Samotné zpracování je rozděleno na tři postupné odevzdání, ve kterých budeme zpřesňovat konečnou podobu aplikace.

Příklad výstupu vašeho programu pro část ulice Domažlická, tj. zároveň vstupu pro systém Graphviz, je zde.

```

strict graph {
  node [ label="", height=0.1, width=0.1 ]

  21311329 [ pos="14.4641165,50.0874984!" ]
  1131753606 [ pos="14.4640847,50.0875633!" ]
  25973241 [ pos="14.4637957,50.0881533!" ]
  25973242 [ pos="14.4635274,50.0887613!" ]
  1131753673 [ pos="14.4633355,50.0892057!" ]
  25973237 [ pos="14.4633016,50.0892842!" ]

  21311329 -- 1131753606
  1131753606 -- 25973241
  25973241 -- 25973242
  25973242 -- 1131753673
  1131753673 -- 25973237
}

```

Překlad ze zdrojového souboru pro Graphviz do PDF se provede programem `neato`:

```
neato -Tpdf -o VYSTUPNI_SOUBOR.pdf VSTUPNI_SOUBOR.gv
```

## Způsob odevzdání

Odevzdávejte jako zdrojové soubory v jazyce Ruby. Všechny potřebné soubory umístěte do adresáře `prijmeni_jmeno_N`, kde N označuje fázi odevzdávání (1/2/3). Tento adresář celý zabalte do souboru `prijmeni_jmeno_N.zip` nebo `prijmeni_jmeno_N.7z` a připojte je postupně jako přílohy k artefaktu s vaším úkolem. Pokud budete používat jakoukoliv knihovnu mimo standardní, musíte ji připojit k odevzdání a popsat její instalaci.

Jako **název odevzdávaného úkolu** vyplňte `GAL DU1`, `GAL DU2`, resp. `GAL DU3` (viz info níže).

Jednotlivé kroky odpovídají třem postupným odevzdáním projektu. Úkoly jednotlivých kontrolních bodů jsou následující:

1. Načíst soubor s mapou ve formátu OSM, vyfiltrovat pouze hrany potřebné pro konstrukci uliční sítě (tudiž například ne okraje budov) a zkonstruovat neorientovaný graf reprezentující danou síť (prozatím budeme ignorovat možnost jednoho směru). Uložte také informaci o délce každého segmentu ulice a povolené rychlosti – pokud informace o povolené rychlosti nebude dostupná, pak ji nastavte jako 50 km/h. Poté zobrazte výsledek načtení do textového výstupu pro program Graphviz. Výstup by měl využívat geografické informace (lat a lon) a mapa ulic by tak měla být dobře poznatelná. Body křižovatek je možné zobrazit výrazněji. Program bude mít následující rozhraní:

```
ruby osm2graphviz.rb --load <map.osm> <output.gv>
```

kde `<map.osm>` značí soubor s mapou a `<output.gv>` soubor, do kterého se zapíše graf ve formátu Graphviz.

2. Program bude sloužit pro zajištění strategie pluzení silnic při velkých sněhových kalamitách. V těchto situacích bude požadováno, aby zástupci města nejprve v daném úseku pročistili silnice, které umožní všem co nejrychlejší jízdu všem. Pro tyto účely byl vybrána jako první síť ulic reprezentována minimální kostrou grafu. Tuto kostru je ovšem nutné vytvořit ze sítě uvažující pouze křižovatkové a koncové body ulic s odpovídajícími hranami mezi nimi. Definujte vážení hran tak, aby odpovídalo skutečné délce segmentů ulic mezi křižovatkami, a vypočtete minimální kostru. Tuto kostru zobrazte do připravené mapy odlišnou barvou a tloušťkou čáry. Pro tyto účely uvažujte prozatím neorientovaný graf. Program bude mít následující rozhraní:

```
ruby osm2graphviz.rb --mst <map.osm> <output.gv>
```

kde `<map.osm>` značí soubor s mapou a `<output.gv>` soubor, do kterého se zapíše graf ve formátu Graphviz.

3. Program bude mít dále schopnost určit pro dvě zadaná místa nejkratší cestu, pro níž také vypíše informaci o době, za jakou vozidlo cestu ujede (za použití maximální povolené rychlosti). Tuto úlohu vypracujte jednak pro neorientovaný a jednak pro orientovaný graf, který bude brát v potaz také případné jednosměrky. Navíc je úkolem tuto cestu zakreslit do mapy odlišnou barvou a tloušťkou čáry a tak zvýraznit její průběh v již připraveném výstupu typu Graphviz. Program bude mít následující rozhraní:

```
ruby osm2graphviz.rb --mindist <map.osm> <lat_start> <lon_start> <lat_stop> <lon_stop>
<output.gv>
```






kde `<map.osm>` značí soubor s mapou, další čtyři parametry souřadnice startovního a cílového bodu a `<output.gv>` soubor, do kterého se zapíše graf ve formátu Graphviz. Vzdálenost a dobu program vypíše na standardní výstup.


## Jazyk Ruby

Jazyk Ruby je preferovaný jazyk odevzdání, který nabízí pro tento typ úlohy vhodné prostředky.

Nicméně, pokud o to opravdu stojíte, můžete úlohu odevzdat také v jazycích Matlab, R, Java nebo Python. V tomto případě ale budu úlohu hodnotit přísněji, vyžadují opravdu **kvalitní komentáře a dokumentaci** a odevzdání musí být bez jakýchkoli dodatečných starostí a stahování, spustitelné se standardními instalacemi těchto jazyků. Pokud odevzdání nebude funkční, nebudu kód podrobně studovat a hledat, co jste v něm náhodou napsali dobře.

## Potřebé zdroje

-  **Základy objektově orientovaného programování**
-  **Formát pro výměnu dat XML**
-  **Grafové algoritmy**
  -  **GAL Graf a jeho reprezentace**
  -  **GAL Průchod grafem**

-  [GAL Hledání minimální kostry](#)
- [www.openstreetmap.org](http://www.openstreetmap.org)
  - Wiki stránky projektu  
[wiki.openstreetmap.org](http://wiki.openstreetmap.org)
  - OSM XML format  
[wiki.openstreetmap.org/wiki/OSM\\_XML](http://wiki.openstreetmap.org/wiki/OSM_XML)
- [www.graphviz.org](http://www.graphviz.org)
  - Příklad souboru pro Graphviz (pro příkaz neato) je v příloze tohoto artefaktu.
  - Překlad zdrojového souboru do PDF se provede příkazem  

```
neato -Tpdf -o VYSTUPNI_SOUBOR.pdf VSTUPNI_SOUBOR.gv
```
- Tutorial  
[graphs.grevian.org](http://graphs.grevian.org)

## Řešení

Domácí úkol odešlete k ohodnocení tlačítkem níže. Dodržujte pokyny pro vyplnění požadavku, například formát přílohy či název požadavku, pokud vyučující takové pokyny uvedl v zadání domácího úkolu.


Odeslat řešení domácího úkolu

## Jak odevzdat domácí úkol

**Vyplňte "Název"** - někteří vyučující uvádějí požadovaný název v zadání domácího úkolu. (*Jméno odesílatele a předmět jsou doplněny automaticky školním systémem.*)

Pokud chcete do požadavku **vkładat přílohy**, klikněte na tlačítko "Vložit přílohu". Zobrazí se dialogové okno, kde zadáte cestu k příslušnému souboru. Po kliknutí na tlačítko "OK" se zobrazí seznam příloh, následně zavřete dialogové okno.

**Domácí úkol odevzdáte kliknutím na tlačítko "Odeslat"**. V záhlaví se zobrazí lišta s informací "Požadavek byl úspěšně odeslán ke zpracování." *Založené ale neodeslané požadavky se hlásí ve vašem úkolovníku aktivitou "Neodesláno!" O akceptaci (či zamítnutí) vyučujícím Vás bude systém informovat aktivitou a požadavek s domácím úkolem se následně odkáže na Vaší kartu studenta v předmětu.*

Podrobná nápověda k odevzdání domácího  [Jak odevzdat domácí úkol](#)

## Zdroje

- [Odkaz na přednášky, které citují potřebné zdroje. Kapitola je citována z příslušné lektorské verze příkladu.](#)