

Letný semester

Zhrnutie práce:

V letnom semestri som sa zameril predovšetkým na optimalizáciu svojho algoritmu a na zavedenie podpory pre prvky relácií (relations) v XML štruktúre.

Optimalizácia:

- Namiesto načítavania celého DOM dokumentu do pamäte som implementoval prúdové (streamové) čítanie XML súboru pomocou StAX. Tento prístup zaručuje konštantnú pamäťovú náročnosť.
- **Časová zložitosť** (vzhľadom na počet prvkov v XML súbore):
 - Bolo: Jeden prechod celým XML súborom pri inicializácii (n) a jeden prechod pri samotnom filtrovaní (n), pričom uzly (node) sa navyše vyhľadávali v množine potrebných ($\log n$). Celková asymptotická zložitosť: $O(n \log n)$
 - Stalo: Jeden prechod celým XML súborom pri inicializácii (n). Jeden prechod pri filtrovaní (n), pričom v množine potrebných prvkov vyhľadávajú už nielen uzly (node), ale aj cesty (way) ($\log n$). Lebo pri sekvenčnom prechode zhora nadol nedokážeme vopred zistiť, či daná cesta zodpovedá požadovaným kritériám (táto informácia sa nachádza v XML nižšie). Pričom o zachovaní alebo zahodení prvku way musíme rozhodnúť ihneď. Celková asymptotická zložitosť: $O(n \log n)$.
- **Priestorová zložitosť**:
 - Bolo: V pamäti bol načítaný celý XML súbor - $O(n)$.
 - Stalo: Do pamäte sa ukladajú iba tie prvky, ktoré skutočne potrebujeme. Asymptotický ide stále o zložitosť $O(n)$, avšak v praxi to predstavuje výrazné zlepšenie.

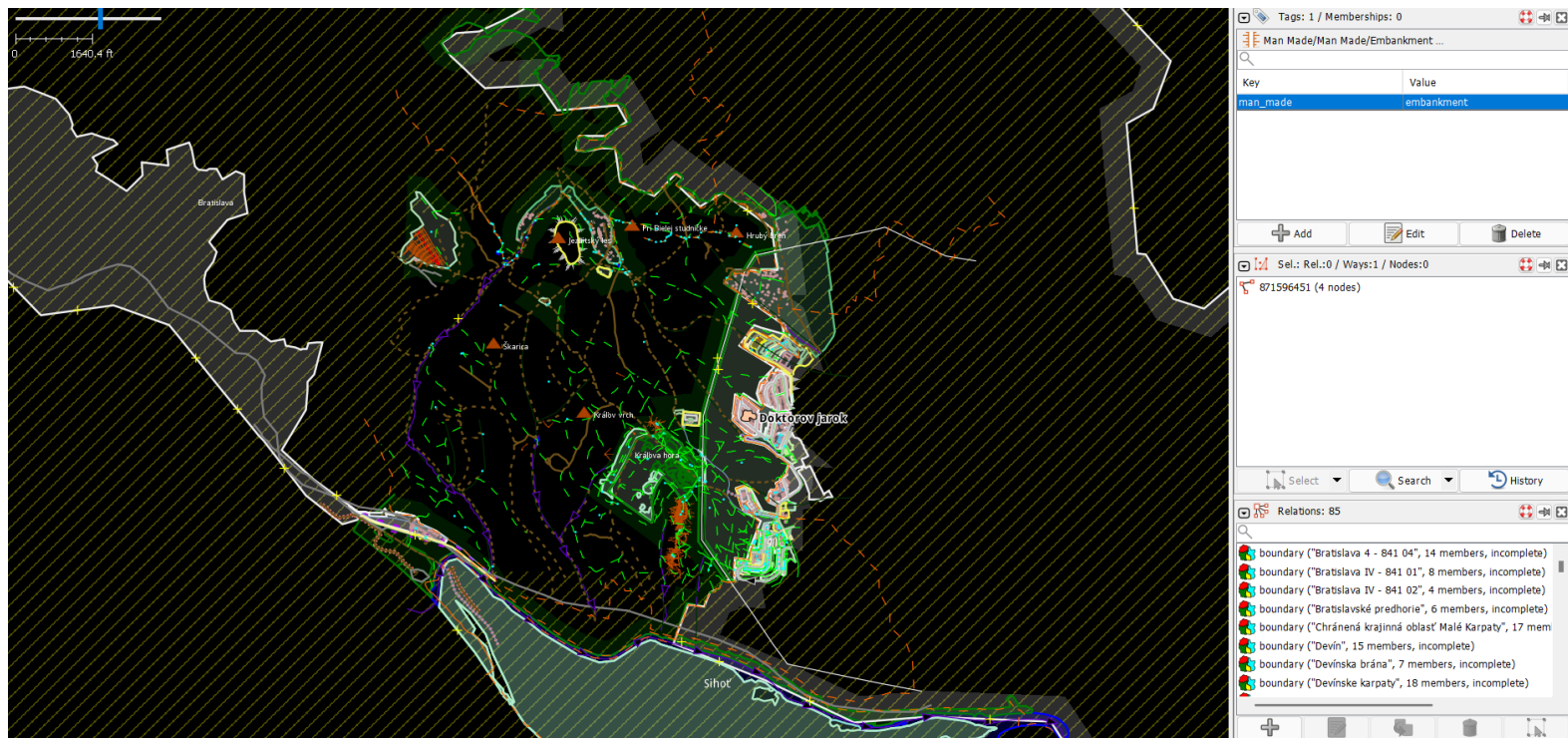
Podpora relácií (vzt'ahov):

Pridal som možnosť spustiť program s príznakom `extended=true`. V tomto režime algoritmus zachová aj relácie (relations), ktoré majú vo výslednej mape aspoň jeden prítomný podradený prvok (dieťa).

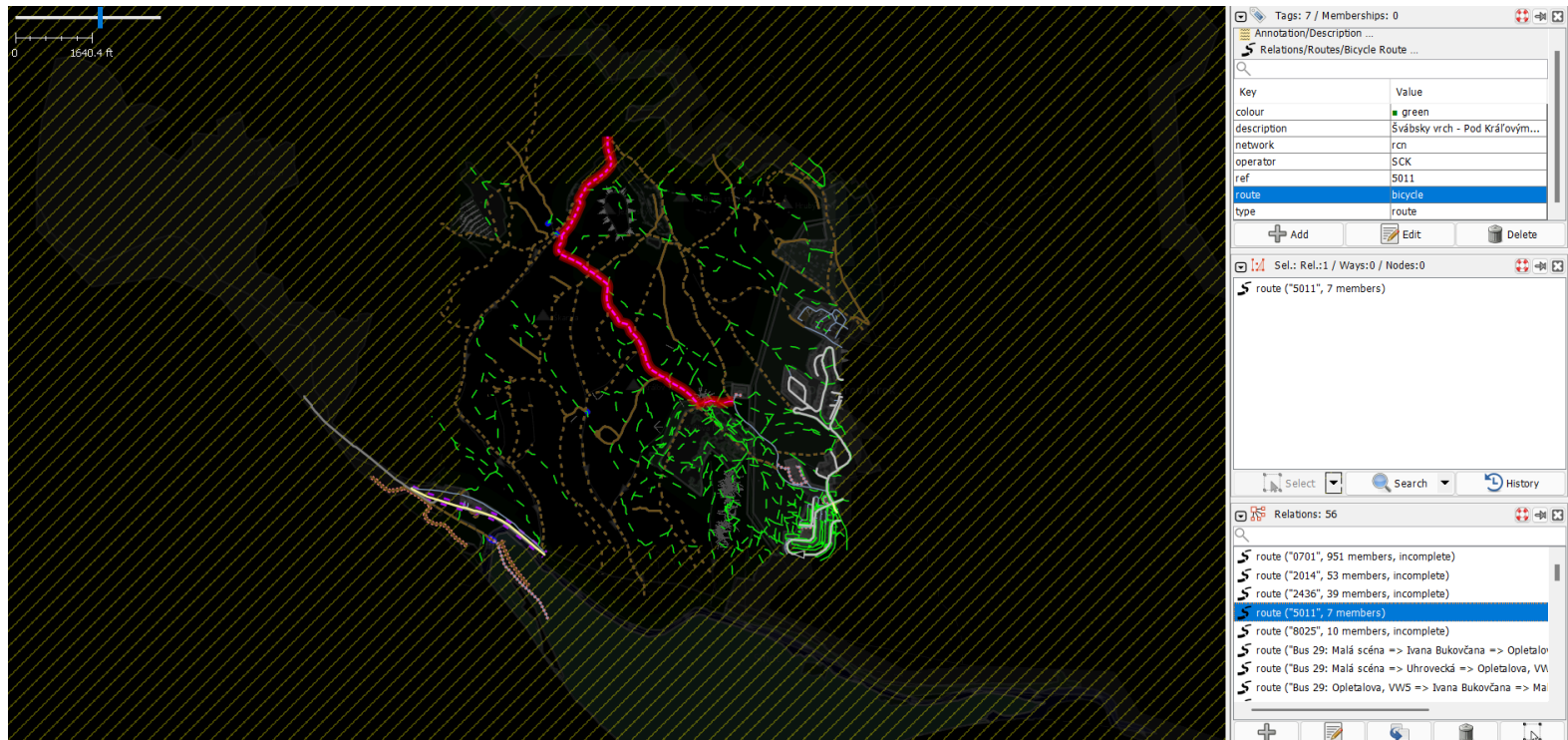
- **Motivácia**: Je vysoko pravdepodobné, že používateľ bude chcieť zachovať informácie o trasách – napríklad o cyklotrasách, trasách pre autá a podobne.
- **Dodatočná časová zložitosť**: Pre každú reláciu (n) program pre každé dieťa overuje, či sa nachádza v množine potrebných prvkov ($\log n$). Môže nastať aj prípad, kedy je podradeným prvkom relácie iná relácia. Program preto musí opakovane vyhľadávať nové relácie dovtedy, kým v poslednom cykle pribudli nejaké nové prvky (čo môže znamenať najviac n prechodov cyklom navyše). Potom celková zložitosť v extended móde: $O(n^2 \log n)$.
- **Dodatočná priestorová zložitosť**: $O(n)$ (v realite podstatne menej).

Příklad použitia

Vstupna mapa:



Výstupná mapa (súčasný program, extended mode):



Vypis terminálu pri predošlej implementácii:

```
Input map name: mountains
Output map name: prev_m

Done. Roads-only OSM created.

Used Memory: 41.32 MB
Total time running: 3.46 seconds

Process finished with exit code 0
```

Vypis terminalu pri súčasnej implementácii v obyčajnom mode:

```
Input map name: mountains
Output map name: curr_m
Keep relevant relations? [Y / N]
N
Collecting data...

Total XML size (in elements): 14807. Nodes: 13419. Ways: 1303. Relations: 85.
Found 4034 nodes belonging to roads.
Found 509 ways that are roads.

Writing filtered OSM file...
Done. Roads-only OSM created.

Used Memory: 3.41 MB
Total time running: 1.96 seconds

Process finished with exit code 0
```

Vypis terminálu pri súčasnej implementácii v extended mode:

```
Input map name: mountains
Output map name: curr_m_relations
Keep relevant relations? [Y / N]
Y
Collecting data...

Total XML size (in elements): 14807. Nodes: 13419. Ways: 1303. Relations: 85.
Found 4034 nodes belonging to roads.
Found 509 ways that are roads.
Found 56 relations having at least one member in the filtered map.

Writing filtered OSM file...
Done. Roads-only OSM created.

Used Memory: 3.45 MB
Total time running: 2.82 seconds

Process finished with exit code 0
```