Térinformatika (Ulyssys)
1. forduló

Ismertető a feladathoz

A feladatok megkezdése előtt győződj meg róla, hogy futnak a következő szoftverek:

- QGIS
- Node.js
- Docker
- DBeaver vagy pgAdmin (vagy más adatbázis eszköz)

Javasolt a következő Docker image-k előzetes letöltése:

```
docker pull postgis/postgis
docker pull fegyi001/oitm-gdal-version
```

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 60 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 1 pont

Hol járunk?

Töltsd le és futtasd a következő projektet: https://github.com/fegyi001/oitm_openlayers

A http://localhost:1234 oldalon futó térkép melyik magyarországi tavunkra nagyít?

Szükséges szoftver: *Node.js, webböngésző*

○ Kis-Balaton

● Tatai Öreg-tó

○ Kiskörei víztározó

○ Orfűi-tó

Magyarázat a megoldáshoz

A Git repository-ból le kell tölteni (git clone vagy zip letöltés), majd ki kell csomagolni (zip letöltés esetén) a forráskódot. A Git repository nyitólapján szerepelnek a függőségeinek telepítésére és a projekt futtatására vonatkozó parancsok.

A térkép a betöltést követően a megoldásban szereplő tóra nagyít.

2. feladat 0 / 1 pont

4269 lépés Magyarországon

Adott kiindulási koordinátákból (19.05315, 47.49855), irányszögből (35°) és távolságból (4629 m) határozzuk meg a beérkezési koordinátákat. A válasz koordinátapár lon, lat sorrendben, pontosvesszővel elválasztva, szóközök nélkül, 6 tizedesre kerekítve fogadható el, tizedesvesszőnek pontot használva (pl. 19.123456;46.123456).

Tipp: a feladat megoldásához nem elengedhetetlen, de amennyiben szeretnél PostGIS-t használni, a legegyszerűbb a következő parancsot futtatni terminál ablakban:

```
docker run -p 5432:5432 --rm -it -e POSTGRES_PASSWORD=postgres postgis/p
```

Mely hatására Docker segítségével letöltésre, majd elindításra kerül egy PostGIS adatbázis a következő kapcsolati paraméterekkel (javasolt DBeaver vagy pgAdmin használata):

- `host: localhost`
- `port: 5432`
- `database: postgres`
- `user: postgres`
- `password: postgres`

A megoldások:
19.088412;47.532650
19.088484;47.532615

Magyarázat a megoldáshoz

Legegyszerűbben PostGIS lekérdezéssel kapunk eredményt.

```
SELECT ST_AsText(ST_Project(ST_GeomFromText('POINT(19.05315 47.49
```

3. feladat 0 / 1 pont

Verzió kereső

A GDAL az egyik legszélesebb körben használt nyílt forráskódú parancssoros térinformatikai programkönyvtár. Harározd meg a *fegyi001/oitm-gdal-version* Docker image-ben található GDAL **verziószámát!**

Tipp: a következő parancs futtatásával a szükséges Docker image letöltésre, majd elindításra kerül, és a futó Docker container terminál ablakjába jutsz:

```
docker run --rm -it fegyi001/oitm-gdal-version bash
```

○ GDAL 2.4.2, released 2019/06/28

○ GDAL 2.4.4, released 2020/01/08

○ GDAL 3.0.4, released 2020/01/28

● GDAL 3.1.2, released 2020/07/07

Magyarázat a megoldáshoz

Az összes GDAL parancs vissza tudja adni a GDAL verziószámát, itt pl. a gdalinfo:

```
gdalinfo --version
```

Melyre a következő válasz érkezik: GDAL 3.1.2, released 2020/07/07

4. feladat 0 / 1 pont

Sentinel-2 csatornaszám

Hány spektrális sávon gyűjt adatokat a Sentinel-2 műholdrendszer MultiSpectral Instrument szenzora?

A megoldás:
13

Magyarázat a megoldáshoz

A Sentinel-2 Multispectral Instrument specifikációiból, de akár Wikipedia-ról is ki lehet olvasni a spektrális sávok számát, azzal a nehezítéssel, hogy a sávok 1 és 12 között vannak számozva, viszont külön 8 és 8A sáv létezik.

5. feladat 0 / 1 pont

Találd meg a vizet!

Sentinel-2 felvételeket adatforrásnak használva az alábbi indexek közül melyik az, amelyik a vízzel borított felületeket emeli ki?

Tippek:

- Amennyiben QGIS-ben szeretnéd megoldani a feladatot, a mintaterület képanyagát mellékletben találod.
- Lehetőség van a Sentinel Hub EO Browser-ében is megoldani a feladatot, ebben az esetben az elinduláshoz szükséges linket a mellékletek között találod (Sentinel_hub)

● `(B03−B08)/(B03+B08)`

○ `(B02−B21/B02+B21)`

○ `(B08−B12)/(B08+B12)`

Magyarázat a megoldáshoz

A (B02-B21/B02+B21) megoldás elvetendő, mert a Sentinel-2 műholdnak nincs 21-es sávja.

A (B03-B08)/(B03+B08) és a (B08-B12)/(B08+B12) közül a helyes kiválasztását többféle módon is megtehetjük.

1. Amennyiben a mellékelt többcsatornás Sentinel-2 kivágatot használjuk, úgy azt QGIS-ben megnyitva a "Raster/Raster calculator" eszközzel elő tudjuk állítani, és össze tudjuk hasonlítani az indexeket.

2. A Sentinel Hub EO Browser használatával online is el tudjuk végezni a feladatot: akár az "index" részben összekattintjuk (**https://tinyurl.com/oitm-find-water-solution**), vagy "custom script"-ként beírjuk a két képletet (**https://tinyurl.com/oitm-find-water-solution2**), és összehasonlítjuk az eredményt.

Természetesen más, rasztermatematikát támogató programozási nyelv vagy szoftver is ugyanazt a megoldást adja, pl. Numpy/RasterIO.

A (B08-B12)/(B08+B12) megoldás (amely egyébként a Normalized Burn Ratio képlete) alacsony értékeket ad a csupasz talajjal vagy száraz növényzettel fedett felületekre, de magas értékeket ad vissza mind a vizekre, mind a zöld szárazföldi növényzetre. A helyes (B03-B08)/(B03+B08) megoldás magas értéket ad a vizekre, és alacsonyat a szárazföldi területekre a talajfelszín borításától függetlenül.

Vissza a kategóriáimhoz

Térinformatika (Ulyssys)
2. forduló

Ismertető a feladathoz

A feladatok megkezdése előtt győződj meg róla, hogy futnak a következő szoftverek:

- QGIS
- Docker
- DBeaver vagy pgAdmin (vagy más adatbázis eszköz)

Javasolt a következő Docker image előzetes letöltése:

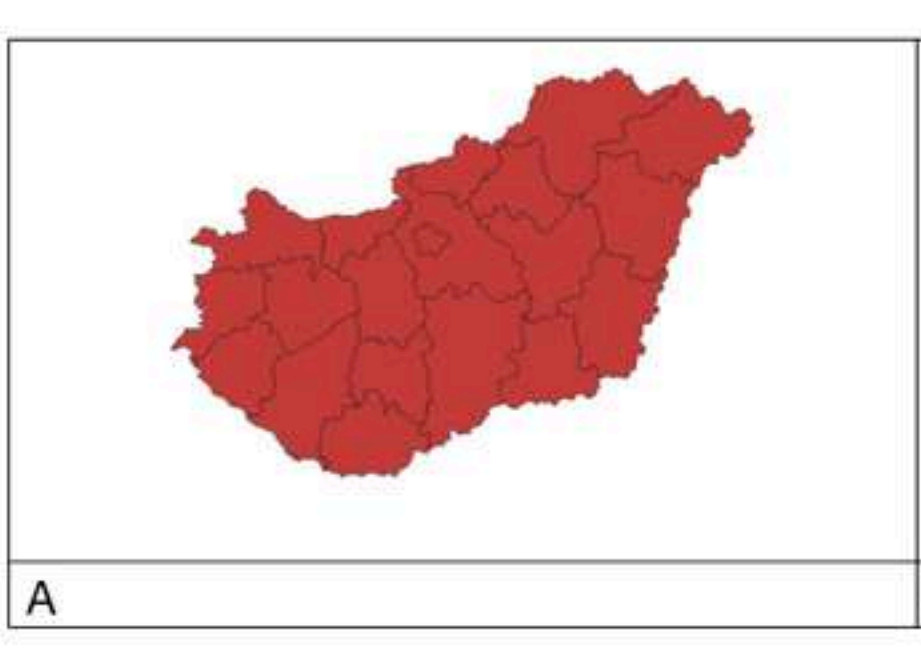
```
docker pull postgis/postgis
```

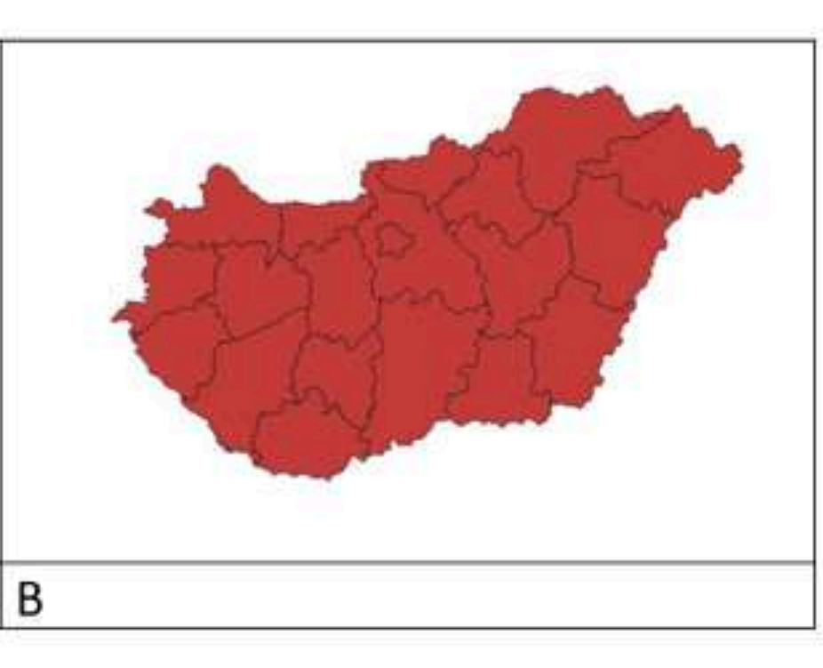
Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 60 perc került rögzítésre mint megoldáshoz felhasznált idő.

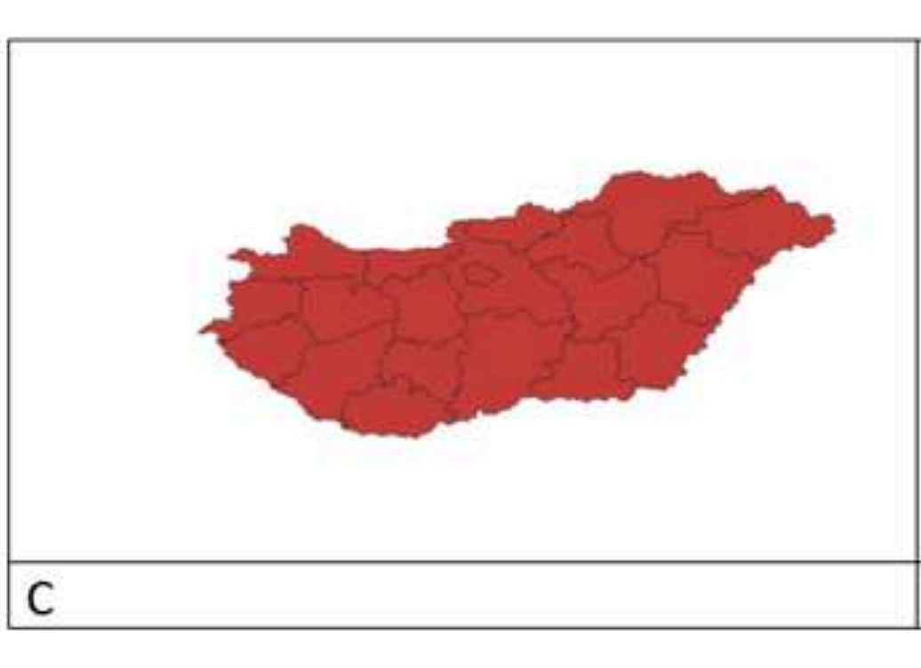
1. feladat 0 / 1 pont

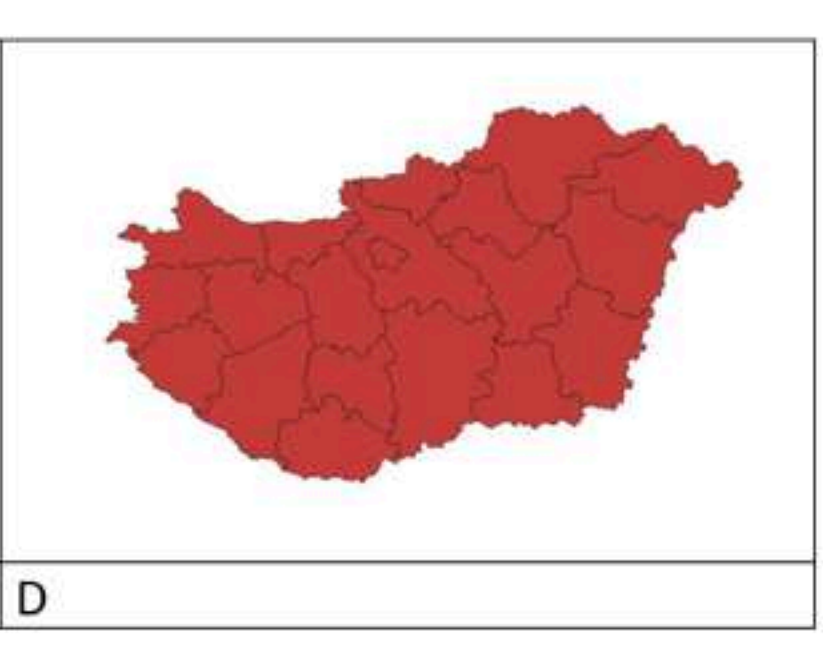
Az alábbi kivágatok közül melyik EOV vetületű?

A kivágatok a következő EPSG kódú vetületi rendszerekben vannak: EPSG:4326, EPSG:23700, EPSG:3035, EPSG:54043.









○ A
● B
○ C
○ D

Magyarázat a megoldáshoz

QGIS-ben Magyarország térképét megnyitva (pl. OpenStreetMap) át lehet állítani a megjelenítési vetületi rendszert a fent felsoroltak mindegyikére, és ez alapján szemrevételezéssel el lehet dönteni, melyik hasonlít legjobban az EOV-ra (EPSG:23700).

2. feladat 0 / 1 pont

A képen látható Python kódrészletben egy többsávos Sentinel-2 fájlból számolunk ki egy ismert indexet. Melyik ez az index?

```
from osgeo import gdal
import numpy

filePath = '/path/to/my/sentinel2_multiband_raster.tif'
img = gdal.Open(filePath)

imgArray = img.ReadAsArray()
bands, rows, cols = imgArray.shape

bandA = img.GetRasterBand(3).ReadAsArray(0, 0, cols, rows)
bandB = img.GetRasterBand(8).ReadAsArray(0, 0, cols, rows)

index = ((bandA-bandB)/(bandA+bandB))

img = None
```

○ NDVI (Normalized Difference Vegetation Index)
○ NDSI (Normalized Difference Snow Index)
○ PSRI (Photochemical Reflectance Index)
● NDWI (Normalized Difference Water Index)

Magyarázat a megoldáshoz

Az NDWI index így számolható ki (forrás: pl. https://en.wikipedia.org/wiki/Normalized_difference_water_index):

```
NDWI = (Xgreen - Xnir) / (Xgreen + Xnir)
```

Sentinel-2 esetében a 3-as csatorna a zöld (Green), a 8-as pedig a közeli infravörös (NIR).

A fenti kódrészletből kiolvasható, hogy ez a két csatorna került felhasználásra az NDWI képletével.

3. feladat 0 / 1 pont

Raszter terepi felbontás

Határozd meg a mellékelt raszter (felbontas.tif) terepi (X és Y) felbontását! Szám érték megadása egy tizedesjegy pontossággal, tizedesvesszőként pontot használva, méterben, X és Y egymástól pontosvesszővel elválasztva, szóközök nélkül (pl. 15.0;-15.0).

A megoldás:
20.1;-20.1

Magyarázat a megoldáshoz

A terepi felbontás több módon is megállapítható.

1. QGIS-be a fájlt behívva, a fájl tulajdonságainak megtekintésekor (Information fül > Pixel Size)

2. GDAL-t használva:

```
gdalinfo felbontas.tif
```

```
Eredményül ezt adja:

Pixel Size = (20.100000000000001,-20.100000000000001)
```

4. feladat 0 / 1 pont

EOV konvertálás

A következő Mercator vetületi rendszerben megadott földrajzi koordinátákat transzformáld át EOV-ba!

- Szélesség: 47.511343°
- Hosszúság: 19.023004°

A választ EOVX;EOVY formátumban kérjük 10 m-re kerekítve (koordináták pontosvesszővel elválasztva, pl. 1230;4560).

Tipp: a feladat megoldásához nem elengedhetetlen, de amennyiben szeretnél PostGIS-t használni, a legegyszerűbb a következő parancsot futtatni terminál ablakban:

```
docker run -p 5432:5432 --rm -it -e POSTGRES_PASSWORD=postgres postgis/p
```

Mely hatására Docker segítségével letöltésre, majd elindításra kerül egy PostGIS adatbázis a következő kapcsolati paraméterekkel (javasolt DBeaver vagy pgAdmin használata):

- `host: localhost`
- `port: 5432`
- `database: postgres`
- `user: postgres`
- `password: postgres`

A megoldások:
240820;648160
648160;240820

Magyarázat a megoldáshoz

A legegyszerűbb PostGIS-ben elvégezni a koordináta transzformációt:

```
select ST_Transform(ST_GeomFromText('POINT(19.023004 47.511343)',
```

Nehezítés, hogy a megoldásban először a második koordinátát kell írni, mert az az EOVX.

5. feladat 0 / 3 pont

Pontok közötti távolság

Add meg a két pont közötti **ellipszoidi** (WGS 84) távolságot méterben 2 tizedesjegy pontosságra kerekítve, tizedesjegynek pontot használva!

- A) pont (Mátyás templom): 47.50202° 19.03428°
- B) pont (Millenáris irodaház): 47.511346° 19.023004°

Tipp: a feladat megoldásához nem elengedhetetlen, de amennyiben szeretnél PostGIS-t használni, a legegyszerűbb a következő parancsot futtatni terminál ablakban:

```
docker run -p 5432:5432 --rm -it -e POSTGRES_PASSWORD=postgres postgis/p
```

Mely hatására Docker segítségével letöltésre, majd elindításra kerül egy PostGIS adatbázis a következő kapcsolati paraméterekkel (javasolt DBeaver vagy pgAdmin használata):

- `host: localhost`
- `port: 5432`
- `database: postgres`
- `user: postgres`
- `password: postgres`

A megoldások:
1340.41
1340.42

Magyarázat a megoldáshoz

Az ellipszoidi távolságot PostGIS geography adattípussal lehet pontosan kiszámítani:

```
with iroda as (
            select ST_GeographyFromText('SRID=4326;POINT(19.02300
                ),
    templom as (
            select ST_GeographyFromText('SRID=4326;POINT(19.03428
    )
select ST_Distance(t.geog, i.geog) from iroda i, templom t;
```

```
Ezen kívül a feladat megoldható sokféleképpen.

Pl. QGIS-ben pont rétegként felvenni a két koordinátát, majd snap

eszközzel összekötni a kettőt, és kiszámolni a távolságot.
```

Vissza a kategóriáimhoz

Térinformatika (Ulyssys)
3. forduló

Ismertető a feladathoz

A feladatok megkezdése előtt győződj meg róla, hogy futnak a következő szoftverek:

- QGIS
- Docker

Javasolt a következő Docker image-k előzetes letöltése:

```
docker pull fegyi001/oitm-geoserver
docker pull postgis/postgis
```

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 60 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 1 pont

Mit mutat a GeoServer?

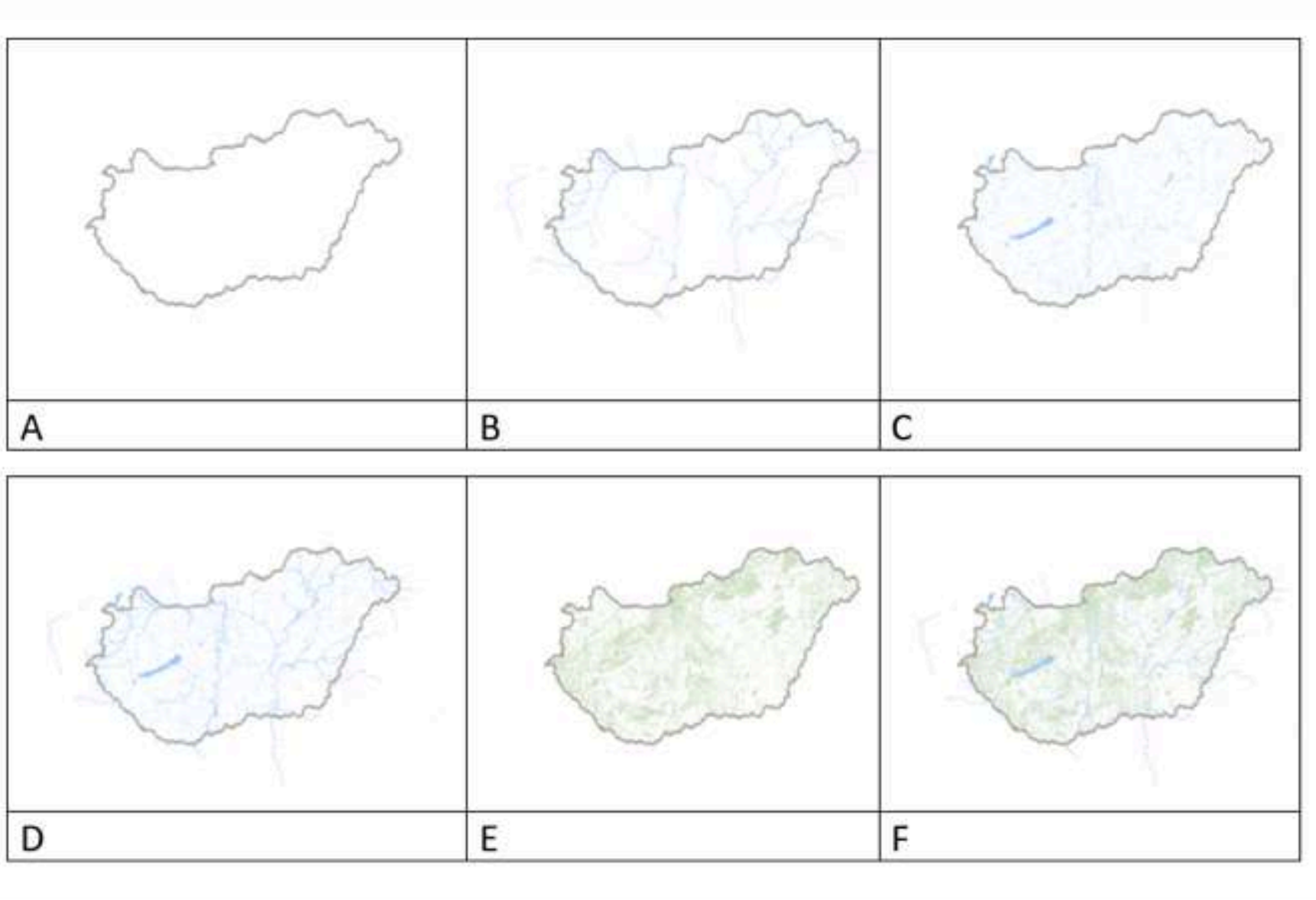
A feladatban a GeoServer-rel, a talán legismertebb térképszerverrel kapcsolatban teszünk fel kérdést. Szükségünk lesz ehhez egy futó GeoServer-re a gépeden, amit Docker segítségével tölts le és indíts a következő módon:

```
docker run -p 8080:8080 -it --rm fegyi001/oitm-geoserver
```

A parancs következtében letöltésre, majd elindításra kerül egy előre konfigurált GeoServer a 8080-as porton (amennyiben épp másra használod a 8080-at, úgy a "8080:8080" rész első tagját cseréld le egy nem használt portra, pl. "8081:8080").

Sikeres indulást követően a http://localhost:8080/geoserver oldalon (vagy 8081 vagy más, amit beállítottál) találod a futó GeoServer-t. Amennyiben be szeretnél lépni az admin felületre, az alapértelmezett név/jelszó párossal tudod azt megtenni.

Nyisd meg előnézeti képen az egyetlen rétegcsoportot! Milyen kép fogad?

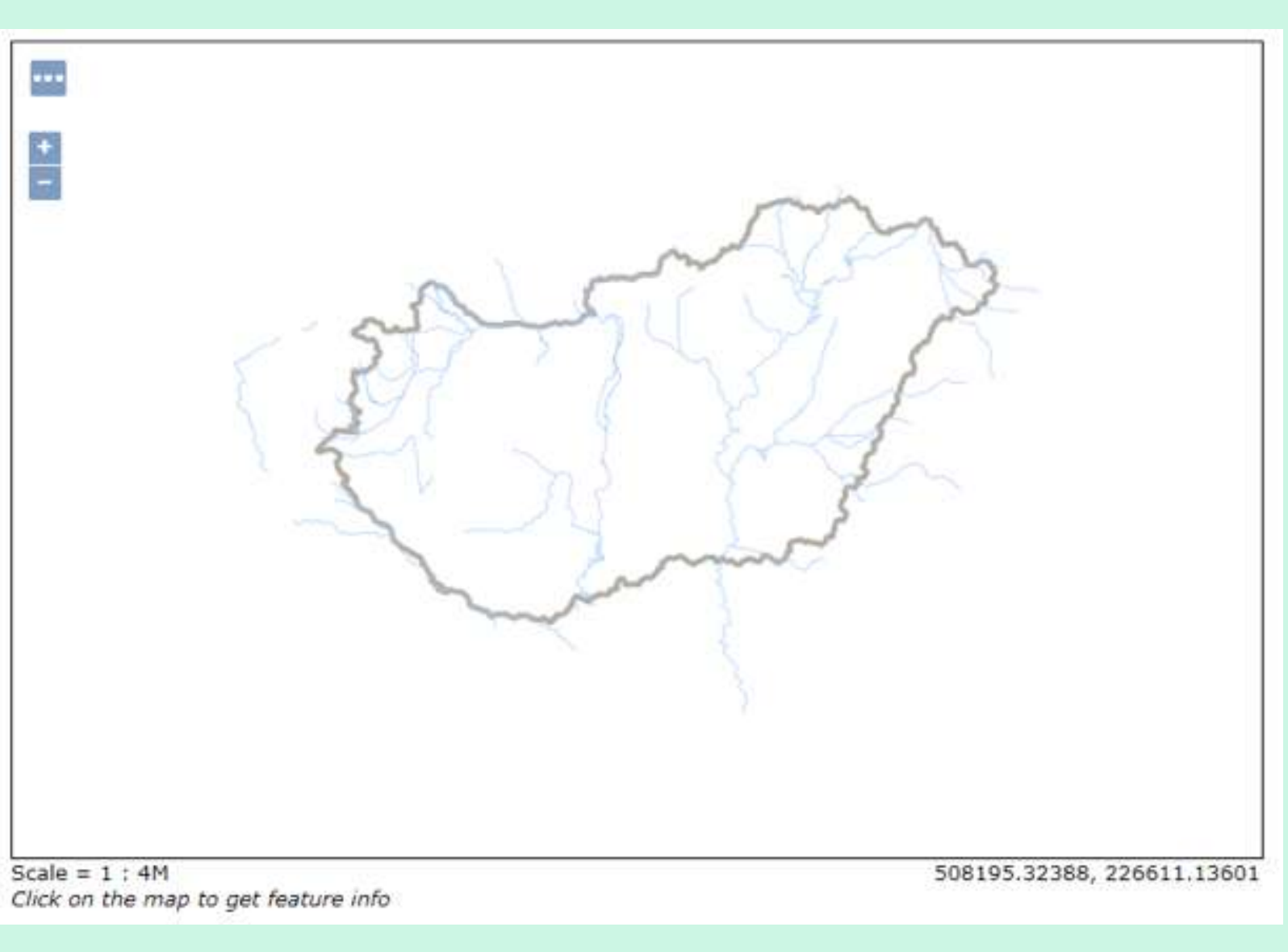


- ○ A (országhatár)
- ● B (országhatár, vízfolyások)
- ○ C (országhatár, tavak)
- ○ D (országhatár, vízfolyások, tavak)
- ○ E (országhatár, erdők)
- ○ F (országhatár, vízfolyások, tavak, erdők)

Magyarázat a megoldáshoz

Nem szükséges az admin felületre való belépés, de a GeoServer dokumentációból könnyen kikereshető, hogy "admin" a felhasználó, és "geoserver" a jelszó. A felület bal oldalán található "Layer preview" bejelentkezés nélkül is elérhető.

Az előre konfigurált GeoServer-ben mindössze két réteg került feltöltésre, egy folyóvíz és egy országhatár réteg. A két réteg mellett egyetlen rétegcsoport található, a "Rivers in Hungary", mely megnyitható itt: http://localhost:8080/geoserver/oitm/wms?service=WMS&version=1.1.0&request=GetMap&layers=oitm%3Amy_group&bbox=365044.39%2C-22800.53%2C967962.66%2C370588.28&width=768&height=501&srs=EPSG%3A23700&format=application/op



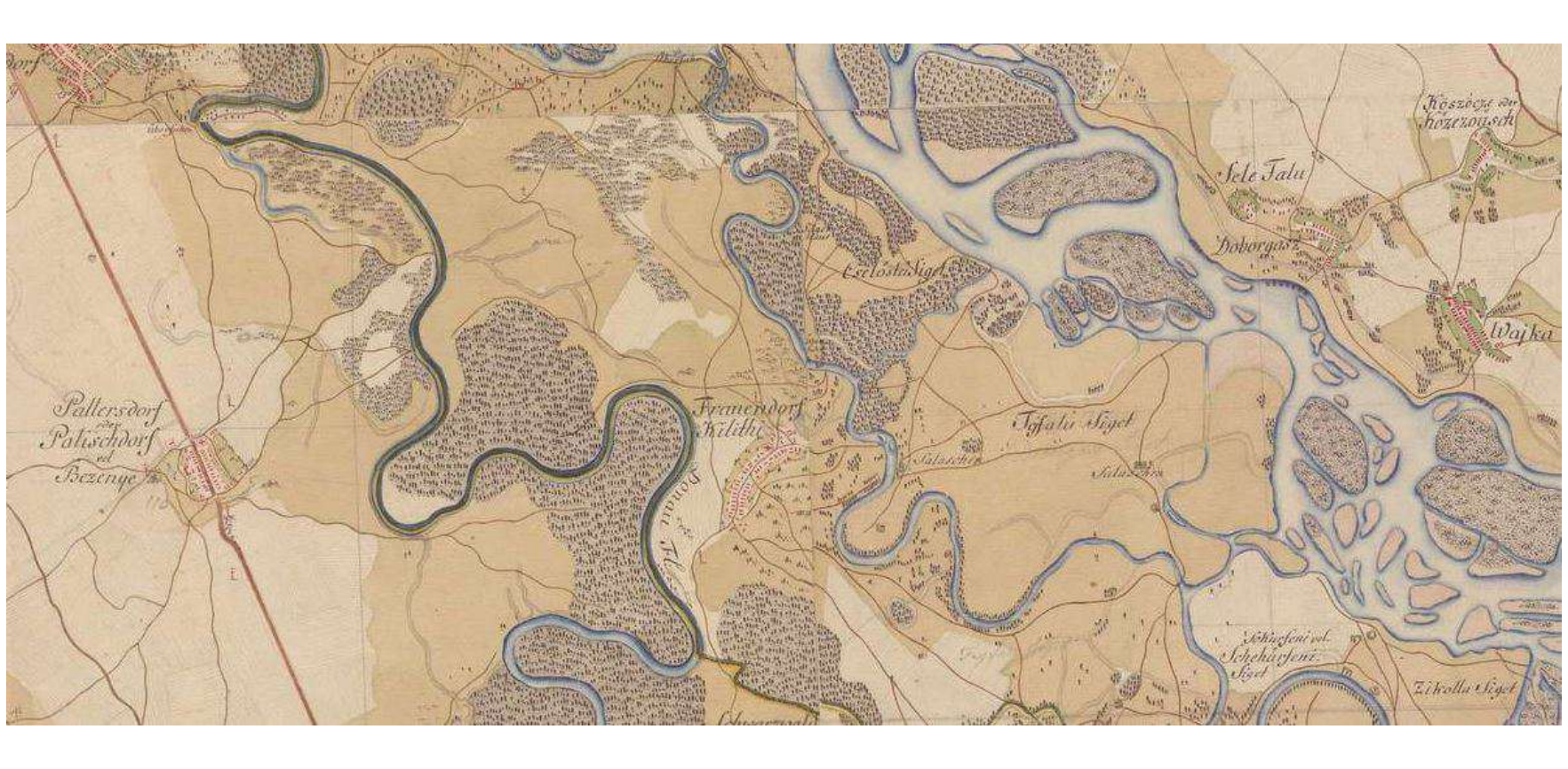
2. feladat 0 / 1 pont

Méretarány az I. katonai felmérésen

Magyarország I. katonai felmérése a 18. század végéről az első olyan térképmű, amelyik településszint alatt ábrázolja közel a teljes országot. Tegyük fel, hogy ebből a térképből a kezünkben fogunk egy szelvényt. Arra vagyunk kíváncsiak, hogy milyen hosszúságú vonallal ábrázolható ezen a szelvényen a valóságban 5739 méter (hány centiméter a térképen)?

A választ centiméterben, két tizedesjegyre kerekítve kérjük (tizedesvesszőként pontot használva), pl. 50.13

Egy kivágat az I. katonai felmérésből, mert olyan szép:



A megoldás:
19.93

Magyarázat a megoldáshoz

Az I. katonai felmérés felmérési méretaránya 1:28 800. (Google találatból kideríthető)

Arányszámítás térképen 1 cm a valóságban 28 800 cm-t jelent. 573 900 cm esetén a választ a következő arány adja meg: 573900/28800

3. feladat 0 / 3 pont

Megye keresés

Találtunk egy levelet, amire cím helyett egy EOV befoglaló téglalap koordinátáit írták fel. Ez alapján szeretnénk legalább a megfelelő megyei postahivatalnak továbbítani a levelet. Melyik megyét keressük fel?

- EOV min X: 254439
- EOV min Y: 801572
- EOV max X: 347223
- EOV max Y: 937478

Tipp: a feladat megoldásához nem elengedhetetlen, de amennyiben szeretnél PostGIS-t használni, a legegyszerűbb a következő parancsot futtatni terminál ablakban:

```
docker run -p 5432:5432 --rm -it -e POSTGRES_PASSWORD=postgres postgis/p
```

Mely hatására Docker segítségével letöltésre, majd elindításra kerül egy PostGIS adatbázis a következő kapcsolati paraméterekkel (javasolt DBeaver vagy pgAdmin használata):

- `host: localhost`
- `port: 5432`
- `database: postgres`
- `user: postgres`
- `password: postgres`

- ○ Vas megye
- ● Szabolcs-Szatmár-Bereg megye
- ○ Pest megye
- ○ Zala megye
- ○ Baranya megye
- ○ Komárom-Esztergom megye

Magyarázat a megoldáshoz

Számos megoldás létezik a feladat megoldására.

Ha valaki tisztában van az EOV koordináták alsó-felső határaival, akkor akár ránézésre is meg tudja mondani, hogy az "EOV max Y" csak a keleti megyéknél tud ilyen magasra kúszni (900000 fölé), a lehetséges válaszok közül pedig csak egyetlen keleti megye van.

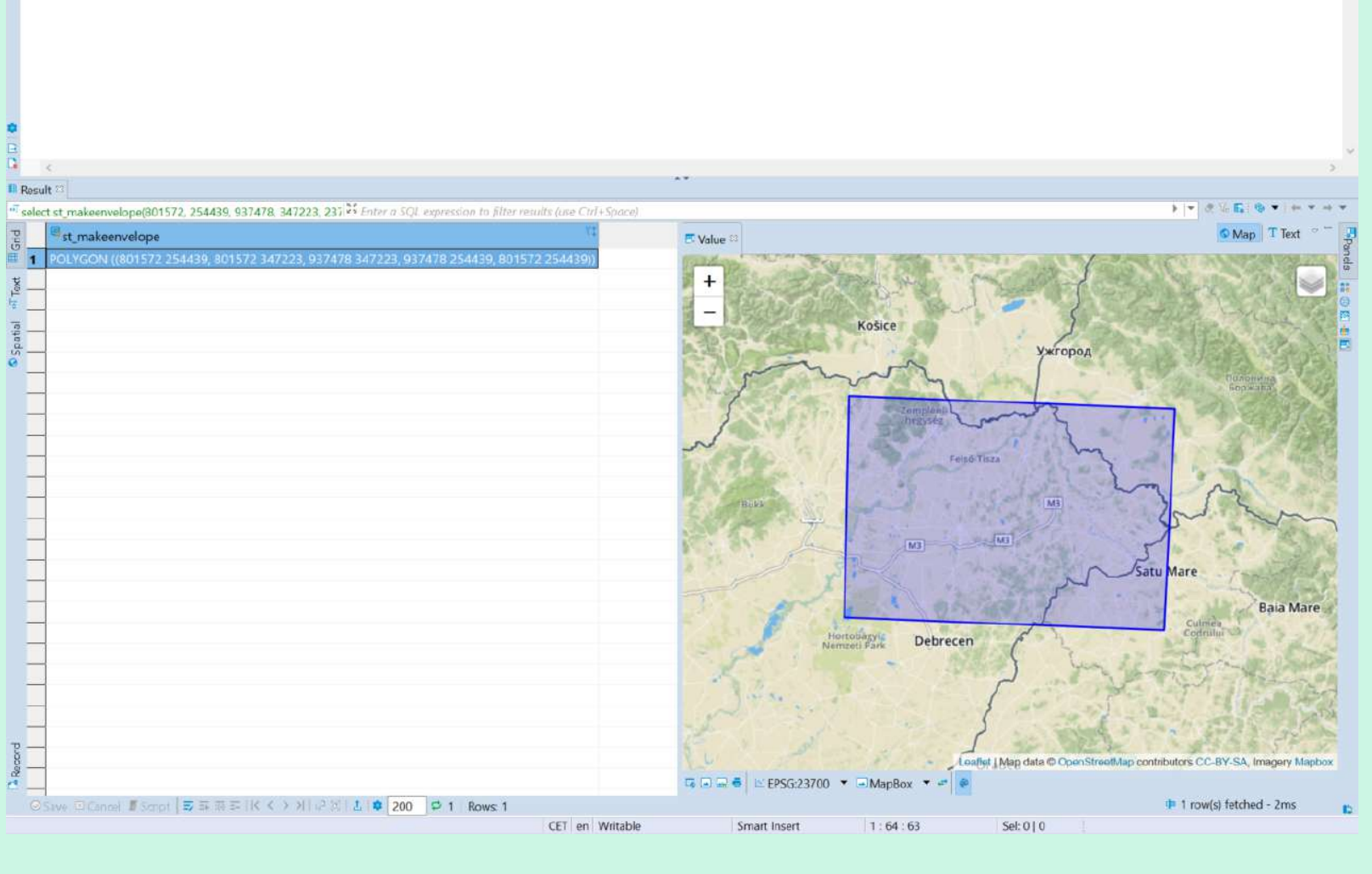
Szoftveresen így oldható meg legkönnyebben: QGIS szoftverben OpenStreetMap térkép megnyitása, vetületi rendszer beállítása EOV-ra, és egérmutató helyzetével meg lehet határozni.

Harmadik megoldás, ha a min és max értékeket átlagolva meghatározzuk a téglalap középpontját: "POINT(869525 300831)", melyet szintén QGIS-ben WKT-ként megjelenítünk a QuickWKT plugin segítségével. Az OpenStreetMap térképpel együtt nézve kiderül, hogy melyik megyében járunk.

Negyedik megoldás, hogy csatlakozunk a PostGIS adatbázishoz, és DBeaver-ben vagy pgAdmin-ban lefuttatjuk a következő SQL-t:

```
SELECT ST_MakeEnvelope(801572, 254439, 937478, 347223, 23700);
```

Mely vizuálisan is megjeleníti az eredményt, melyről azonnal eldönthető, melyik megyére vonatkozik.



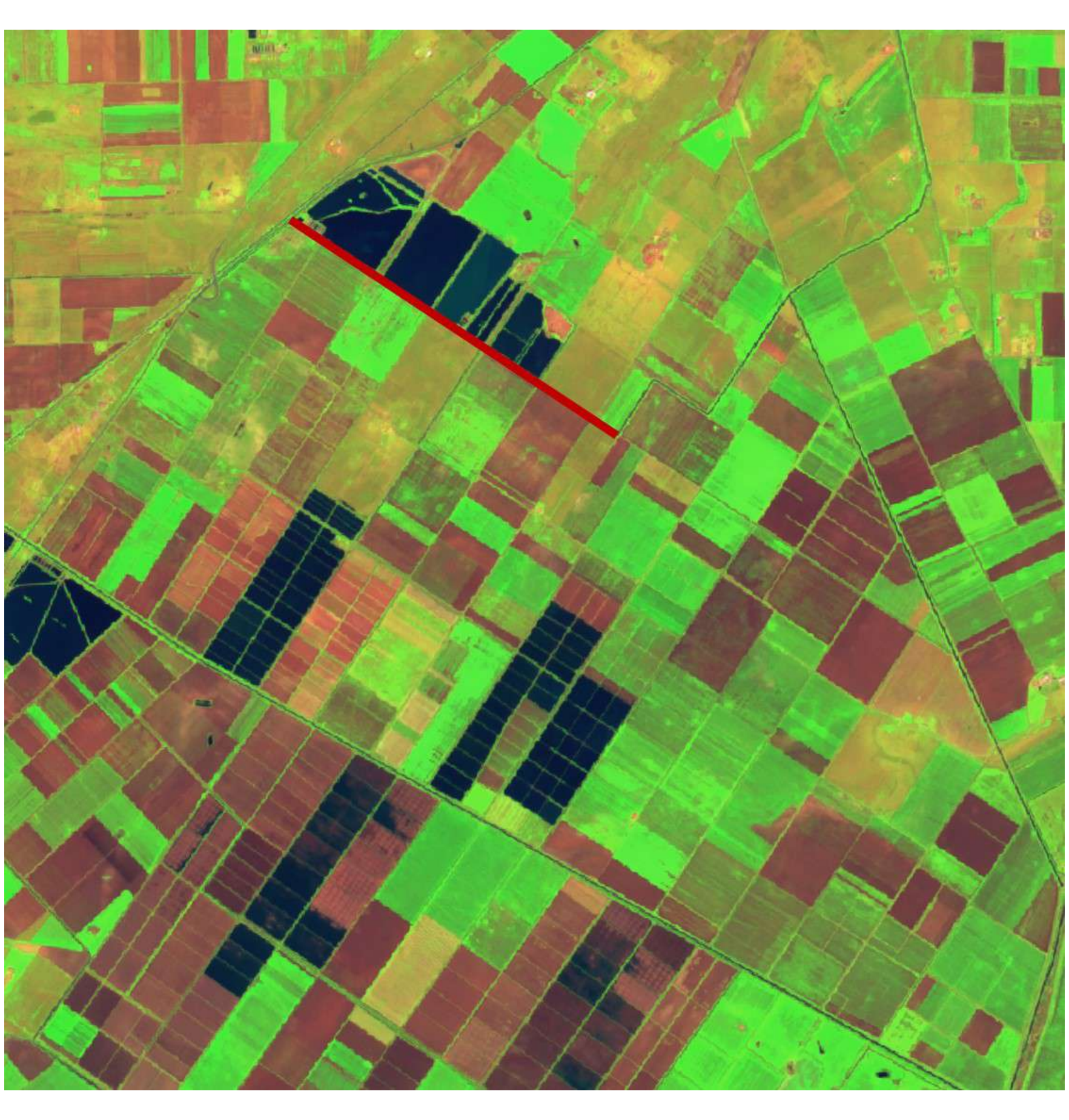
```
select st_makeenvelope(801572, 254439, 937478, 347223, 23700);
```

```
st_makeenvelope
1 POLYGON ((801572 254439, 801572 347223, 937478 347223, 937478 254439, 801572 254439))
```

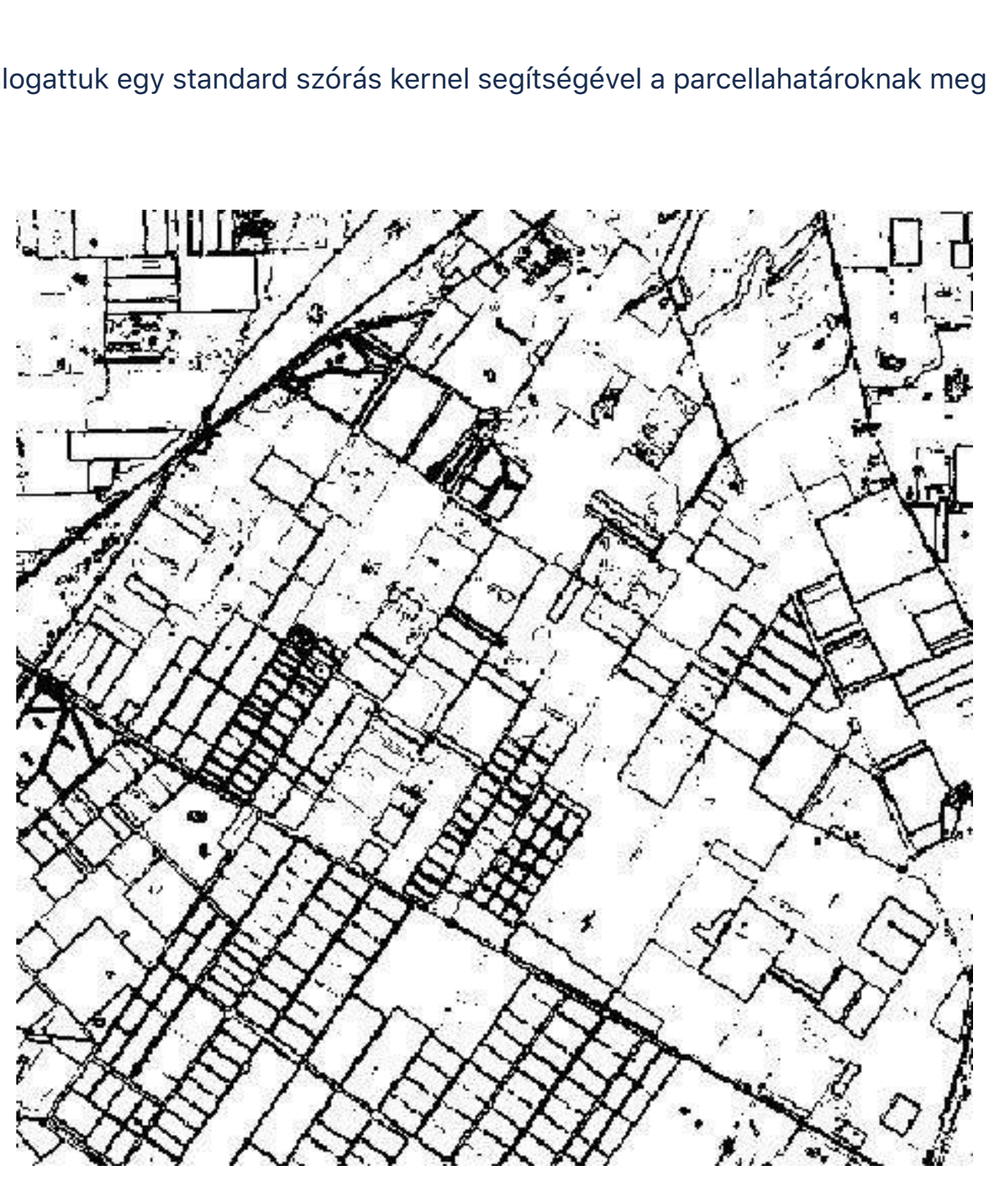
4. feladat 0 / 1 pont

Kernel választó

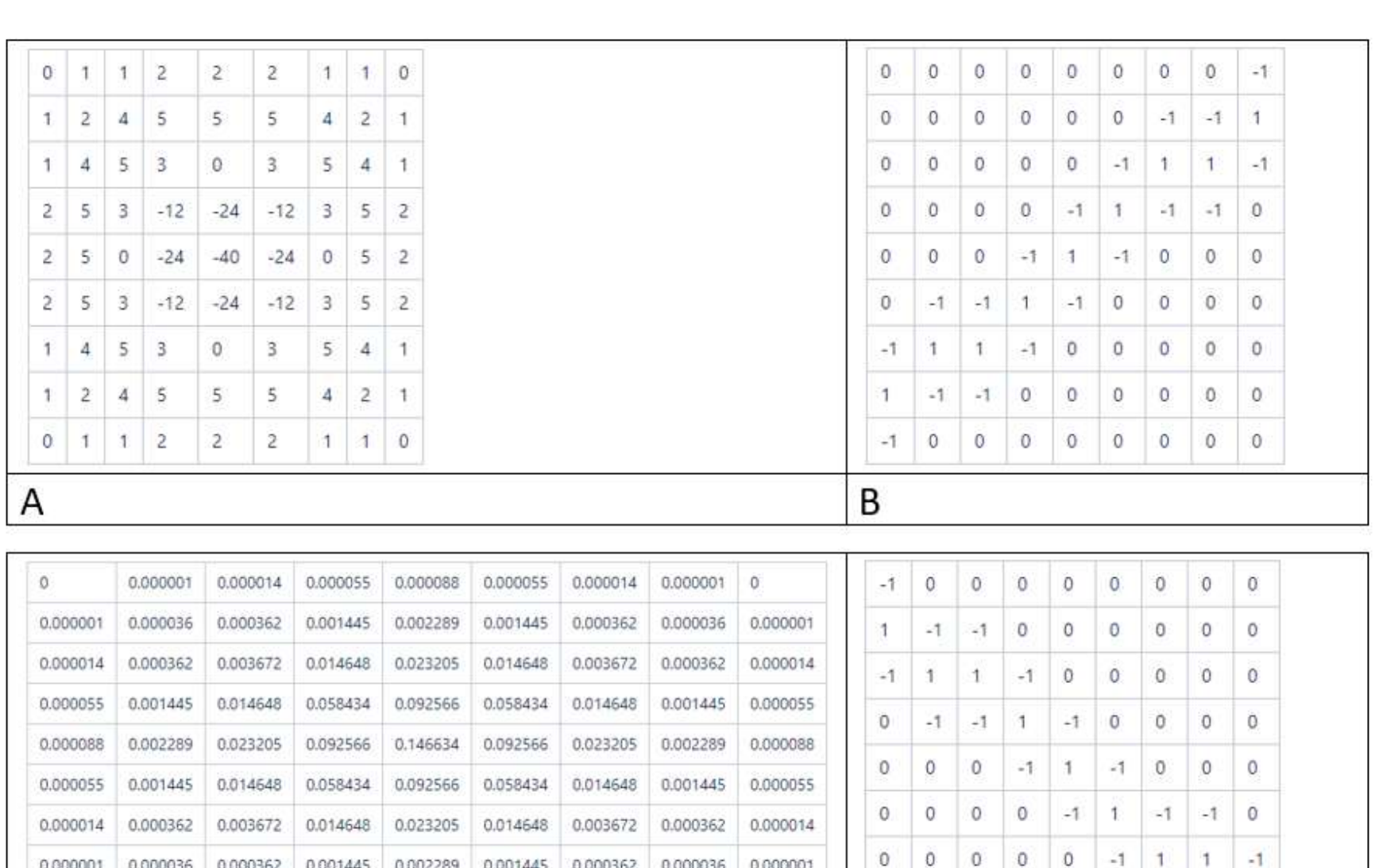
A hortobágyi rizsföldeket ábrázoló műholdképről szeretnénk leválogatni azokat a parcellahatárokat, amelyek a Kecskéri-csatorna piros vonallal megjelölt szakaszával párhuzamosak:



Sikeresen leválogattuk egy standard szórás kernel segítségével a parcellahatároknak megfelelő pixeleket:



Egy kernelszűrési művelet során szeretnénk kiemelni az összes élt, ami párhuzamos a piros vonallal. Melyik kernel lesz a megfelelő? Válaszd ki a megfelelő szűrőt:



A

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

B

0	0	0	0	0	0	0	0	-1
0	0	0	0	0	0	-1	-1	1
0	0	0	0	0	-1	1	1	-1
0	0	0	0	-1	1	-1	-1	0
0	0	0	-1	1	-1	0	0	0
0	-1	-1	1	-1	0	0	0	0
-1	1	1	-1	0	0	0	0	0
1	-1	-1	0	0	0	0	0	0
-1	0	0	0	0	0	0	0	0

C

0	0.000001	0.000014	0.000055	0.000088	0.000055	0.000014	0.000001	0
0.000001	0.000036	0.000362	0.001445	0.002289	0.001445	0.000362	0.000036	0.000001
0.000014	0.000362	0.003672	0.014648	0.023205	0.014648	0.003672	0.000362	0.000014
0.000055	0.001445	0.014648	0.058434	0.092566	0.058434	0.014648	0.001445	0.000055
0.000088	0.002289	0.023205	0.092566	0.146634	0.092566	0.023205	0.002289	0.000088
0.000055	0.001445	0.014648	0.058434	0.092566	0.058434	0.014648	0.001445	0.000055
0.000014	0.000362	0.003672	0.014648	0.023205	0.014648	0.003672	0.000362	0.000014
0.000001	0.000036	0.000362	0.001445	0.002289	0.001445	0.000362	0.000036	0.000001
0	0.000001	0.000014	0.000055	0.000088	0.000055	0.000014	0.000001	0

D

-1	0	0	0	0	0	0	0	0
1	-1	-1	0	0	0	0	0	0
-1	1	1	-1	0	0	0	0	0
0	-1	-1	1	-1	0	0	0	0
0	0	0	-1	1	-1	0	0	0
0	0	0	0	-1	1	-1	-1	0
0	0	0	0	0	-1	1	1	-1
0	0	0	0	0	0	-1	-1	1
0	0	0	0	0	0	0	0	-1

- ○ A
- ○ B
- ○ C
- ● D

Magyarázat a megoldáshoz

Ránézésre is megállapítható, hogy az A és a B kernel mind É-D, mind K-Ny irányban szimmetrikus, csak a C és a D kernel irányszelektív. A C kernelben a nem nulla cellák iránya délnyugat-északkelet, ezeket az irányú éleket emeli ki, a D kernelben a nem nulla cellák iránya északnyugat-délkelet, így ez fogja kiemelni a keresett éleket.

Python-ban ki is próbálható az OpenCV segítségével (opencv-python library).

Vissza a kategóriáimhoz

Térinformatika (Ulyssys)
4. forduló

Ismertető a feladathoz

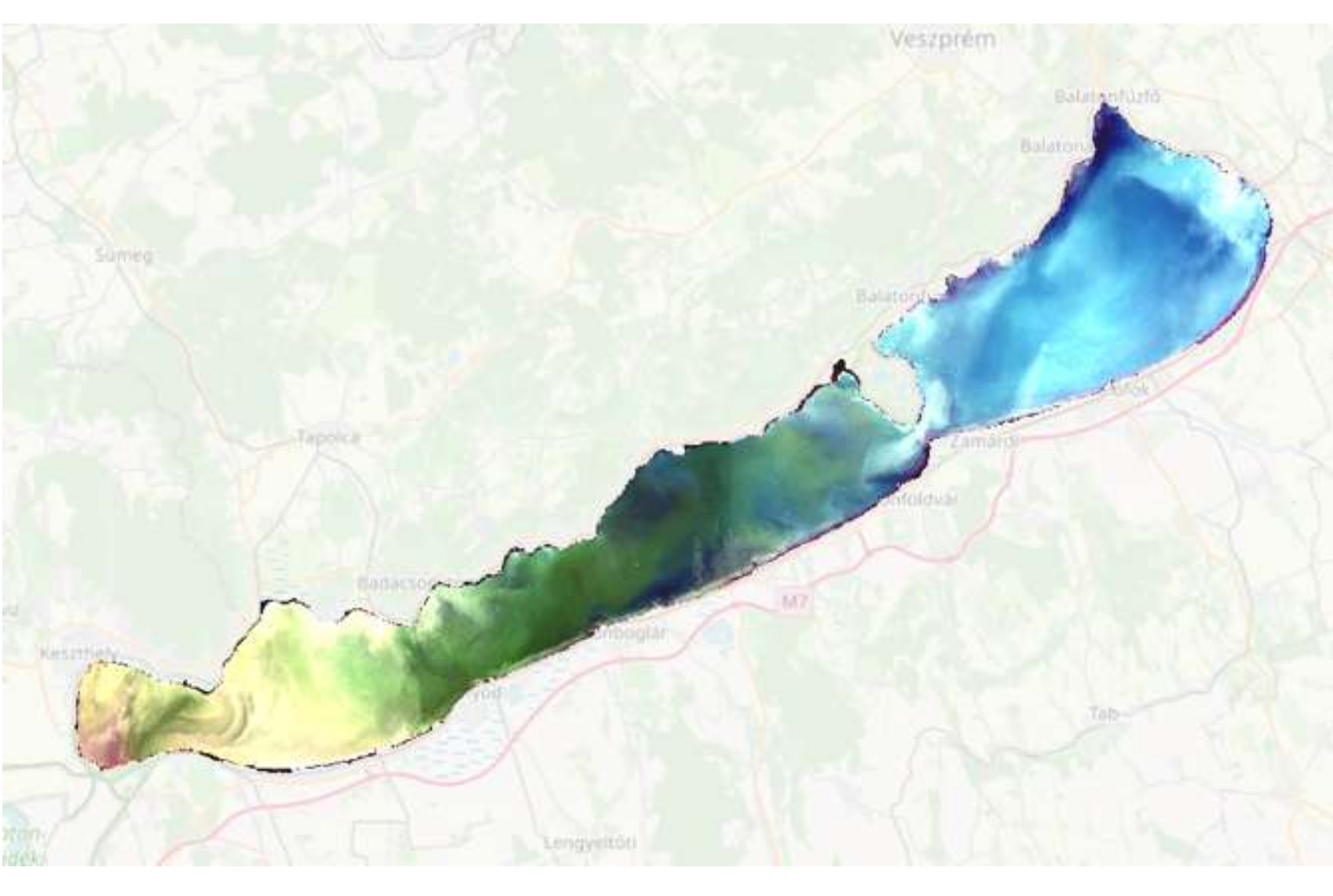
A feladatok megkezdése előtt győződj meg róla, hogy futnak a következő szoftverek:

- QGIS

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 60 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

2020. július 11-én készült egy többcsatornás Sentinel-2 műholdfelvétel, amely jól mutatta a Balaton kezdődő algásodását.



Ahhoz, hogy eldöntsük, hova érdemes menni fürdeni, meg kell állapítanunk, hogy mely településeket érint az algásodás, és hol jó még a vízminőség. Rakd sorrendbe az alábbi Balaton-parti településeket a környékükön vízben lévő algák mennyisége szerint (minél kevesebb az alga, annál előbbre kerüljön a település)!

- Ábrahámhegy
- Alsóörs
- Badacsonytomaj
- Balatonfüred
- Balatongyörök
- Balatonlelle
- Balatonmáriafürdő
- Balatonszárszó
- Balatonudvari
- Fonyód
- Keszthely
- Révfülöp
- Zamárdi
- Zánka

A vízben élő algák mennyiségét a Maximum Chlorophyll Index (MCI) segítségével tudjuk megbecsülni, amely a következő képlettel számolható ki a Sentinel-2 műhold csatornáiból:

```
B05-((0.74-0.705)/(0.74-0.665))*B04-(1.0-(0.74-0.705)/(0.74-0.665))*B06
```

Minél magasabb az index értéke, annál algásabb a víz, annál rosszabb a vízminőség. A .zip file egy shapefile-t tartalmaz a települések neveivel és körzeteivel, valamint egy Sentinel-2 műholdkép kivágatot többcsatornás .tif formátumban. Használd az egyes települések itt megadott körzetein belül az MCI átlagértékét, ennek megfelelően rendezd növekvő sorrendbe a településeket! A megoldásban a településeket vesszővel, majd szóközzel válaszd el egymástól (pl. Zamárdi, Révfülöp, Fonyód).

A megoldások:
Balatonudvari, Alsóörs, Zánka, Balatonfüred, Balatonszárszó, Ábrahámhegy, Zamárdi, Balatonlelle, Badacsonytomaj, Révfülöp, Balatonmáriafürdő, Fonyód, Balatongyörök, Keszthely
Balatonudvari,Alsóörs,Zánka,Balatonfüred,Balatonszárszó,Ábrahámhegy,Zamárdi,Balatonlelle,Badacs

Magyarázat a megoldáshoz

A rasztert QGIS szoftverben betöltve a Raster Calculator eszközzel létre kell hozni egy új, egycsatornás rasztert, amely a Maximum Chlorophyll Index értékeit tartalmazza a megadott képlet szerint. A Processing Toolbox-ban található Zonal Statistics (Raster analysis / Zonal Statistics) eszközzel lehet kiszámítani a MCI átlagértékeit a települések körzetein belül. Ezt követően a települések attribútum tábláját sorba kell rendezni az átlagértékek szerint.

2. feladat 0 / 3 pont

Forgass meg! Ki vagyok?

Rendelkezésünkre áll hazánk egy ismert épülete három dimenziós pontfelhő formájában (lásd mellékelt XYZ fájl). Melyik épület ez?

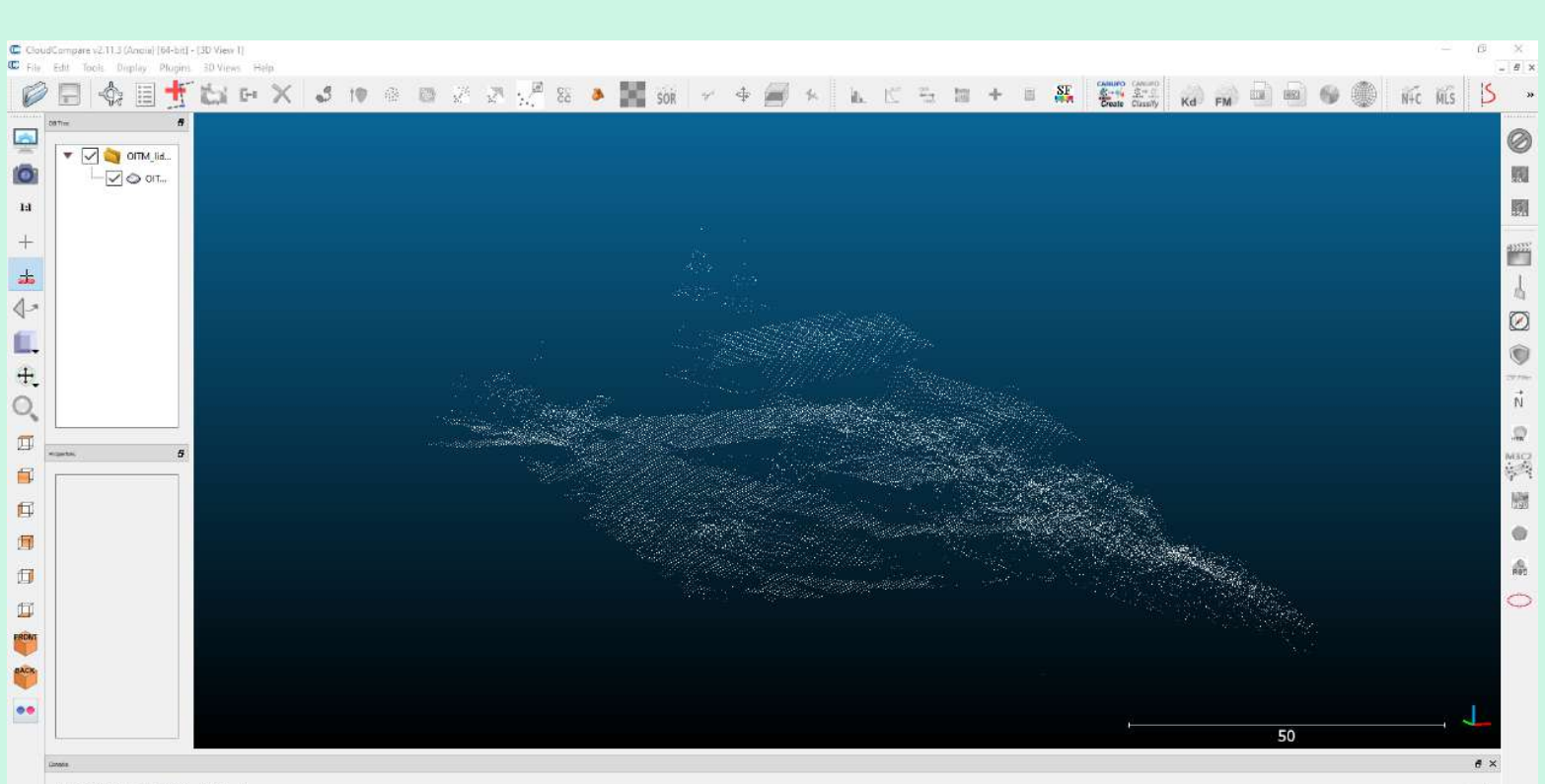
○ Szegedi Dóm

● Tihanyi Apátság

○ Sümegi Vár

Magyarázat a megoldáshoz

Az XYZ fájl megnyitható pl. CloudCompare szoftverben, vagy Python MatPlotLib segítségével. A 3D-s állományt forgatva felismerhetővé válik a Tihanyi Apátság.



3. feladat 0 / 1 pont

Itt a metszés ideje!

Adott két poligon: a millenáris irodaház (kék poligon) és egy fölé rajzolt egyszerű poligon (félig áttetsző piros).

Az alábbi 4 PostGIS lekérdezés közül melyik adja eredményként az irodaház délkeleti szárnyát?

○

```
with iroda as (
select
    ST_GeomFromText('
            Polygon((
                648136 240870,
                648126 240848,
                648190 240771,
                648223 240800,
                648214 240806,
                648196 240792,
                648145 240850,
                648149 240860,
                648136 240870))', 23700) as geom,
    'iroda' as name ),
poligon as (
select
    ST_GeomFromText('
            Polygon((
                648199 240802,
                648216 240815,
                648246 240786,
                648208 240743,
                648175 240781,
                648199 240802))', 23700) as geom,
    'poligon' as name )
select
    ST_Union(i.geom,
    p.geom)
from
    iroda i,
    poligon p;
```

○

```
with iroda as (
select
    ST_GeomFromText('
            Polygon((
                648136 240870,
                648126 240848,
                648190 240771,
                648223 240800,
                648214 240806,
                648196 240792,
                648145 240850,
                648149 240860,
                648136 240870))', 23700) as geom,
    'iroda' as name ),
poligon as (
select
    ST_GeomFromText('
            Polygon((
                648199 240802,
                648216 240815,
                648246 240786,
                648208 240743,
                648175 240781,
                648199 240802))', 23700) as geom,
    'poligon' as name )
select
    ST_Intersects(i.geom,
    p.geom)
from
    iroda i,
    poligon p;
```

●

```
with iroda as (
select
    ST_GeomFromText('
            Polygon((
                648136 240870,
                648126 240848,
                648190 240771,
                648223 240800,
                648214 240806,
                648196 240792,
                648145 240850,
                648149 240860,
                648136 240870))', 23700) as geom,
    'iroda' as name ),
poligon as (
select
    ST_GeomFromText('
            Polygon((
                648199 240802,
                648216 240815,
                648246 240786,
                648208 240743,
                648175 240781,
                648199 240802))', 23700) as geom,
    'poligon' as name )
select
    ST_Intersection(i.geom,
    p.geom)
from
    iroda i,
    poligon p;
```

○

```
with iroda as (
select
    ST_GeomFromText('
            Polygon((
                648136 240870,
                648126 240848,
                648190 240771,
                648223 240800,
                648214 240806,
                648196 240792,
                648145 240850,
                648149 240860,
                648136 240870))', 23700) as geom,
    'iroda' as name ),
poligon as (
select
    ST_GeomFromText('
            Polygon((
                648199 240802,
                648216 240815,
                648246 240786,
                648208 240743,
                648175 240781,
                648199 240802))', 23700) as geom,
    'poligon' as name )
select
    ST_Within(i.geom,
    p.geom)
from
    iroda i,
    poligon p;
```

Magyarázat a megoldáshoz

Nem szükséges PostGIS-t futtatni a feladat megoldásához, mert az SQL-ek szemrevételezésével egyértelműen megállapítható a helyes válasz.

A 3-as lehetőség adja ki a délkeleti szárnyat. Az **ST_Intersection**(a.geom, b.geom) függvény két geometria közös geometriáját adja eredményként.

Az első esetben az **ST_Union**(a.geom, b.geom) összevonja a két geometriát, így visszakapjuk az egész irodaház és a fenti poligon együttes geometriáját.

A második eset (**ST_Intersects**) boolean választ ad a lekérdezés végére.

A negyedik eset (**ST_Within**) azt vizsgálja, hogy egyik objektum teljesen része-e a másiknak. Boolean értéket ad vissza.

Vissza a kategóriáimhoz

Térinformatika (Ulyssys)
5. forduló

Ismertető a feladathoz

A feladatok megkezdése előtt győződj meg róla, hogy futnak a következő szoftverek:

- QGIS
- Docker
- DBeaver vagy pgAdmin (vagy más adatbázis eszköz)

Javasolt a következő Docker image előzetes letöltése:

```
docker pull fegyi001/oitm-postgis-osm
```

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 60 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

Melyik a legmagasabb fa a parkban?

A fák magasságának, állományszerkezetének vizsgálatára gyakran alkalmazott módszer a légi lézerszkennelés (LIDAR). Itt feladat a badacsonyörsi Folly-arborétum legmagasabb fájának azonosítása. A mellékelt .xyz kiterjesztésű fájl tartalmazza a pontfelhőt, a .shp fájl pedig tíz lehetséges legmagasabb fa hozzávetőleges pozícióját. Válaszul a legmagasabb fa azonosítóját kell megjelölni (1-10).

A rendelkezésünkre álló fájlok UTM-33 vetületi rendszerűek.

☑ 1-es azonosítójú fa
☐ 2-es azonosítójú fa
☐ 3-as azonosítójú fa
☑ 4-es azonosítójú fa
☐ 5-ös azonosítójú fa
☐ 6-os azonosítójú fa
☐ 7-es azonosítójú fa
☐ 8-as azonosítójú fa
☐ 9-es azonosítójú fa
☐ 10-es azonosítójú fa

Magyarázat a megoldáshoz

A helyes megoldás az 1 vagy a 4, mindkét megoldást helyesnek fogadjuk el, a feldolgozás módjától függ, hogy melyik fát fogja legmagasabbnak kiadni a rendszer. A legmagasabb fa valójában az 1-es, de a hagyományos, normalizált magasságmodell raszter alapú megoldás a 4-es fát adja legmagasabbnak, kis különbséggel.

A feladat nagyon sokféle módon megoldható, raszteres vagy vektoros alapon, különböző térinformatikai vagy lidar elemző szoftverekben. Itt egy lehetséges megoldási lehetőséget mutatunk be QGIS szoftverben.

- - Az XYZ pontfelhőt importáljuk QGIS szoftverbe 'Add layer' / 'Add delimited text layer eszközzel', kiválasztva értelemszerűen az x,y, és z koordinátákat tartalmazó oszlopokat. A vetület beállításakor olyat keressünk, ahol a koordináták mértékegysége méter.

-Nyissuk meg a shapefilet: ez pontokat jelöl, amelyek az egyes fák hozzávetőleges középpontjai. Ahhoz, hogy az egyes fák magasságát megállapítsuk, a legmagasabb és legalacsonyabb pontok magasságának különbségére van szükségünk ezen pontok környezetében.

-Képezzünk puffert az egyes fákat azonosító pontok körül. A puffer távolságát úgy kell kiválasztani, hogy a legközelebbi fák esetében se fedjen át, tehát a két egymáshoz legközelebbi fa távolságának fele lesz a puffer távolság: 2,5 méter. ('Vector / 'Geoprocessing Tools' / Buffer')

-Egyenként kiválasztva a puffer poligonjait, le tudjuk válogatni a pontfelhőnek a hozzájuk tartozó pontjait külön fájlokba ('Vector' / 'Geoprocessing Tools' / 'Clip')

-Ezeknek a néhány pontból álló fájloknak a legmagasabb és legalacsonyabb pontjai közötti magasságkülönbség megfeleltethető a fa magasságának. Ezeket az értékeket a 'Vector' / 'Analysis Tools' / 'Basic Statistics for Fields' eszközzel tudjuk vizsgálni, sorban kiválasztva az egyes fájlt és azoknak a magassági koordinátáját tartalmazó mezőt. Az eszköz által kiadott statisztikák közül "Range" paraméter adja meg a legmagasabb és legalacsonyabb pont közötti különbséget.

Vissza a kategóriáimhoz

Térinformatika (Ulyssys)
6. forduló

Ismertető a feladathoz

A feladatok megkezdése előtt győződj meg róla, hogy futnak a következő szoftverek:

- QGIS
- Node.js
- Docker

Javasolt a következő Docker image előzetes letöltése:

```
docker pull fegyi001/oitm-complex-geoserver
```

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 60 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

Kristálygömb a tavon

Ebben a feladatban együtt fogod használni az OpenLayers-t és a GeoServer-t. Egy pont típusú shapefájlt szeretnénk betölteni GeoServer-be, és egy webes térképes alkalmazásban megjeleníteni. A konkrét feladat csak a térképes alkalmazás megnyitásakor fog kiderülni!

A feladathoz szükséges kódot tartalmazó Git repository-t töltsd le innen: **https://github.com/fegyi001/oitm_complex_frontend**

A kódbázis tartalmaz egy GeoServer-t is, melyet a következő módon hozz létre (terminál ablakból, a letöltött kódot tartalmazó könyvtárból):

```
# 1. saját GeoServer Docker image létrehozása
docker build -t my-geoserver .

# 2. a most létrehozott GeoServer futtatása 8080-as porton
docker run -p 8080:8080 --rm -it my-geoserver
```

Ha fut a GeoServer, lépj be az alapértelmezett név-jelszó párossal, majd publikáld a GeoServer-en belül található **data/upload** mappa egyetlen shape fájlját úgy, hogy alapértelmezett stílusként a **pointStyle**-t állítod be!

Ha sikeres volt a WMS réteg létrehozása, akkor nincs más hátra: add hozzá az OpenLayers-es alkalmazáshoz legfelső rétegként! (Az alkalmazás indítása a forráskód Readme.md fájljában leírt módon történik.)

Nyisd meg a térképet a http://localhost:4200 -as oldalon, és az ott leírt kérdésre add meg a választ!

Megjegyzés: a projekt Angular keretrendszerben írt webes térképes megjelenítő, de a feladat megoldásához semmilyen Angular ismeret nem szükséges.

○ Piros (#FF0000)

○ Kék (#0000FF)

○ Fehér (#FFFFFF)

● Magenta (#FF00FF)

○ Sárga (#FFFF00)

○ Fekete (#000000)

○ Zöld (#008000)

Magyarázat a megoldáshoz

A GeoServer-re való belépést "admin/geoserver" név/jelszó párossal tehetjük meg, ez az alapértelmezett.

A shapefájl publikálása alapfeladat GeoServer-en, ennek megfelelően jól dokumentált online (https://docs.geoserver.org/latest/en/user/gettingstarted/shapefile-quickstart/index.html): először egy Store-t kell létrehozni, ami a shapefájl elérési útjára mutat, majd a Store-ból egy Layer-t lehet publikálni. A Layer létrehozásakor be kell állítani a Bounding Box-ot, és a Publishing fülön lehet kiválasztani az alapértelmezett stílust legördülőből, ahol kiválaszthatjuk a feladathoz létrehozott pointStyle-t.

A GeoServer WMS réteg létrehozása után a következő módon tudjuk hozzáadni a réteget az app.component.ts fájlban (ez az egyetlen fájl a forráskódban, ahol OpenLayers rétegek és térkép szerepel):

```
new TileLayer({
    source: new TileWMS({
    url: 'http://localhost:8080/geoserver/wms',
    params: {
      LAYERS: ['oitm:point'],
      FORMAT: 'image/png8',
    },
  }),
}),
```

A dolgot egyszerűsíti, hogy ugyanitt már szerepel egy nagyon hasonló WMS réteg. Gyakorlatilag csak duplikálni kell ezt a néhány sor kódot, és átírni benne a réteg nevét.

Mivel a kódban szerepel, hogy 14-es zoom szinttel induljon a térkép, így garantált, hogy a térkép betöltésekor a kör színe mindig magenta lesz.

Az eredmény így néz ki:

A képen elrejtettünk egy színes kristálygömböt. Az a különlegessége, hogy különböző magasságokból nézve képes megváltoztatni a színét.

Milyen a színe a térkép betöltésekor?

2. feladat 0 / 5 pont

Merre folyik a folyó?

Egy sietve végzett munka során véletlenül rossz formátumban tároltunk egy domborzati modellt: raszter helyett xyz pontok formájában. A feladat megoldásához helyre kell állítani a rasztert, és azonosítsd be, hogy a modellen látszó legnagyobb völgyben merre folyik a folyó! A választ szóközzel együtt add meg (pl. keletről nyugatra)!

Mellékletben találod az XYZ fájlt! (folyo.zip)

Megjegyzés: vetületi rendszernek használj méter alapút (pl. UTM-33).

○ északnyugatról délkeletre

● délkeletről északnyugatra

○ délnyugatról északkeletre

○ északkeletről délnyugatra

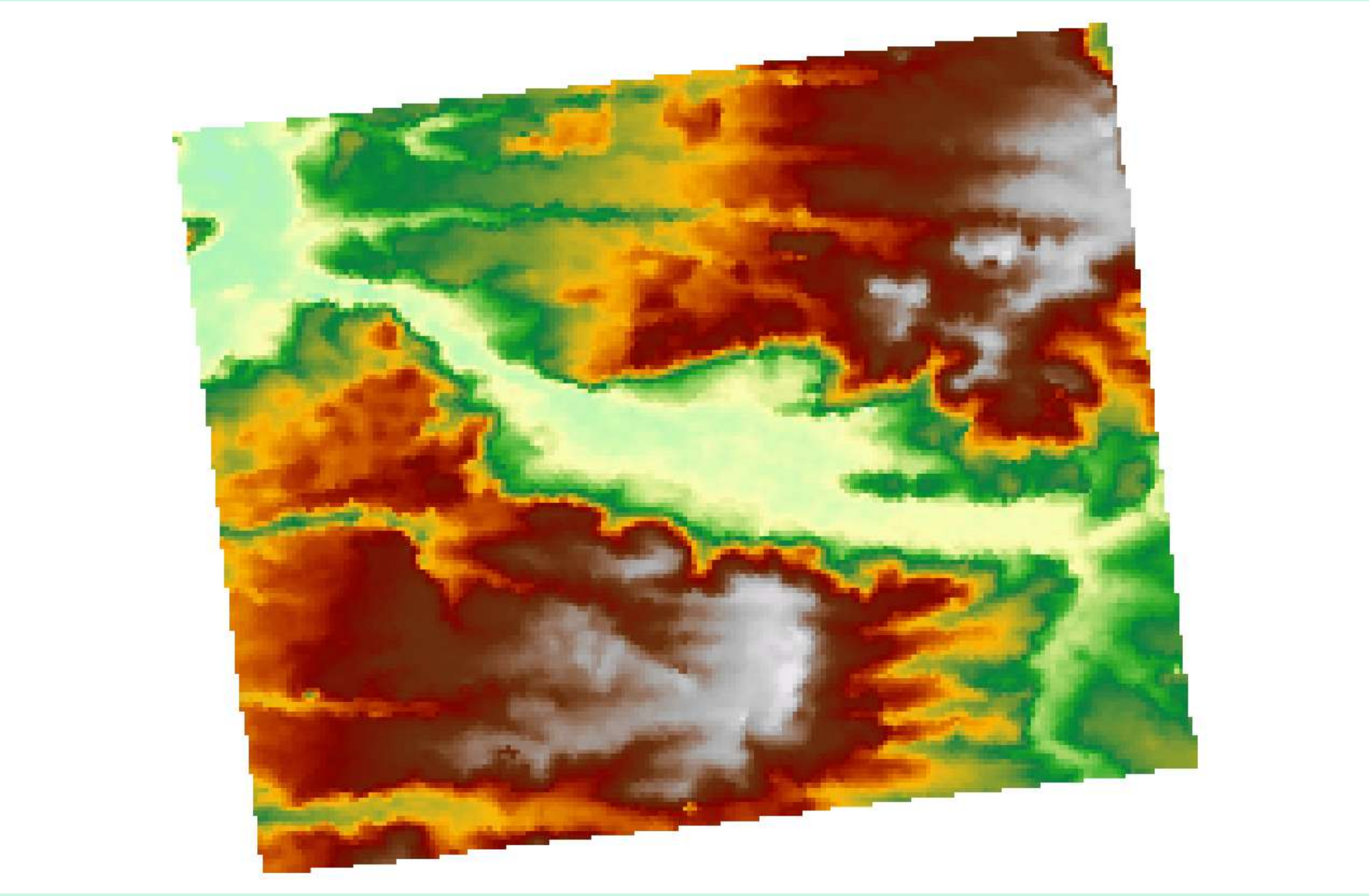
Magyarázat a megoldáshoz

A feladat megoldásának első lépése a raszter-vektor konverzió. Ehhez az xyz formátumú rasztert be kell olvasni QGIS-be a "Layer" / "Add Delimited Text Layer" eszközzel, majd az így előállt réteget el kell menteni egy új fájlba, Esri Shapefile formátumban.

Ez a shapefile lesz a bemenet a "Raster" / "Conversion" / "Rasterize" eszköz számára, ahol a "Field to use for a burn-in value" paraméterben lehet megadni, hogy a shapefile harmadik oszlopa legyen a raszter magassági értéke. Mivel a pontok egymástól való távolsága X és Y irányban nem azonos (20, illetve 30 méter), a két távolság legkisebb közös többszörösét, 60 métert érdemes megadni cellaméretnek.

Az így elkészült magassági modellen már látszik az északnyugat-délkelet irányú folyóvölgy. A lefolyás irányának meghatározásához a magassági modellnek a völgy két végébe eső raszter cellák magasságát kell leolvasni a QGIS Identify eszközzel (vagy olyan paletta szerint kell kiszínezni a réteget, hogy szemrevételezéssel is meg lehessen állapítani a választ). A völgy délkeleti, szélesebb végén 148 méter körüliek a magasságok, míg a völgy északnyugati, keskenyebb végén 140 méter körüliek, a folyó tehát délkeletről északnyugatra folyik.

Az eredmény QGIS-ben megjelenítve:



Vissza a kategóriáimhoz

Térinformatika (Ulyssys)
7. forduló

Ismertető a feladathoz

A feladatok megkezdése előtt győződj meg róla, hogy futnak a következő szoftverek:

- QGIS
- GRASS GIS
- Docker
- DBeaver vagy pgAdmin (vagy más adatbázis eszköz)

Javasolt a következő Docker image előzetes letöltése:

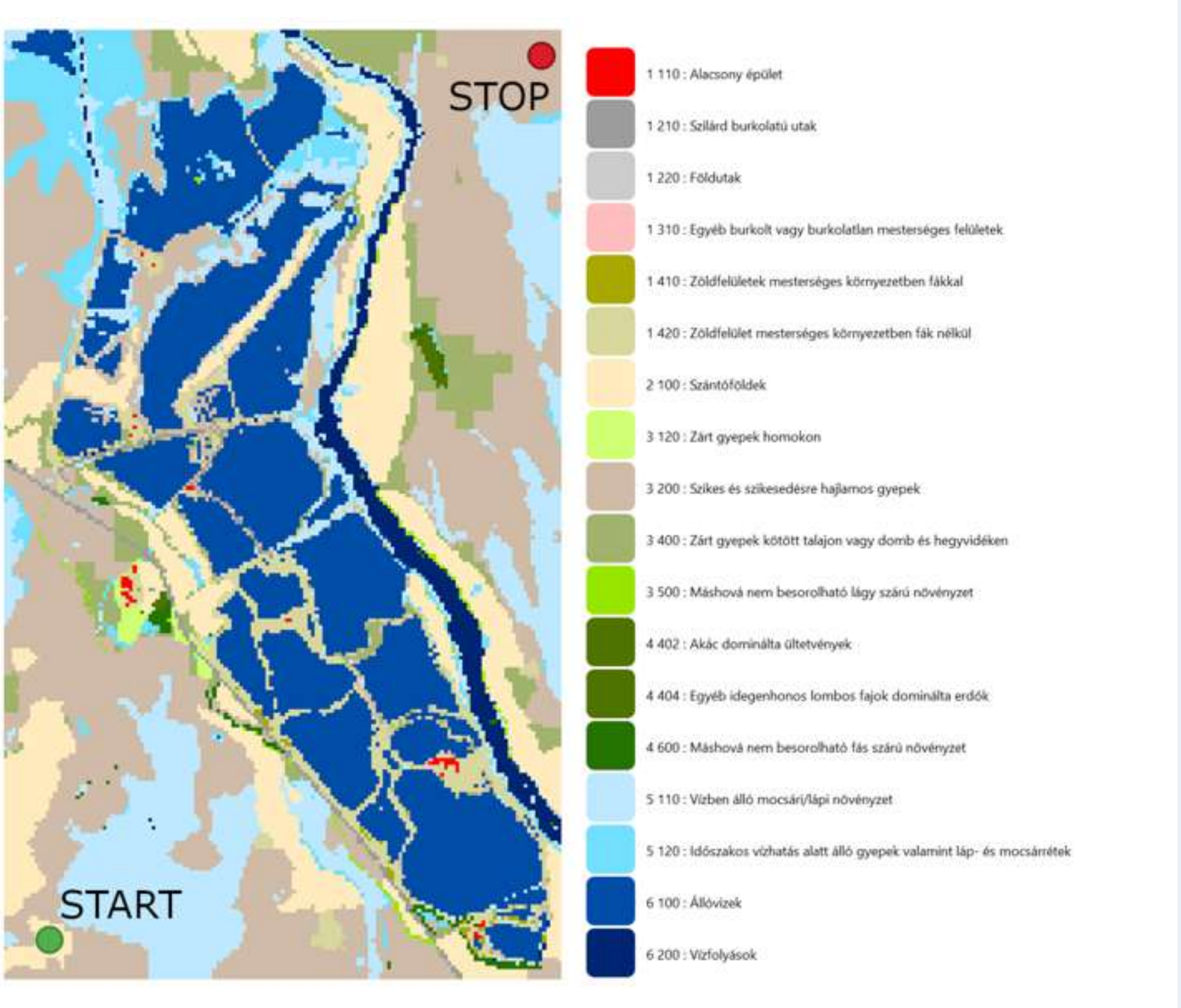
```
docker pull fegyi001/oitm-postgis-osm
```

Tekintettel arra, hogy egy választ sem rögzítettél az alábbi feladatlapon, ebben a fordulóban a kitöltésére rendelkezésre álló idő teljes egésze, azaz 60 perc került rögzítésre mint megoldáshoz felhasznált idő.

1. feladat 0 / 5 pont

Mikor fogytán az erőnk...

A Kiskunsági szikes tavaknál járunk, és nagyon elfáradtunk. Szeretnénk átjutni a tó-hálózaton, de úgy, hogy az a lehető legkevesebb erőkifejtést igényelje. Rendelkezésünkre áll a terület NÖSZTÉP térképe, ami az ökoszisztéma hálózatot mutatja: ez segítségünkre lehet abban, hogy elkerüljük a "drága" (nehezen járható) helyeket, és helyettük az "olcsó" (könnyen járható) területeket részesítsük előnyben. Melyik útvonalon érdemes végighaladnunk? A zöld pont (délnyugati sarok) a START, a piros pont (északkeleti sarok) a STOP.



A NÖSZTÉP térképből kiindulva elkészíthetünk egy ún. költség-térképet, a kategóriák újraosztályozásával (minél magasabb a költség, annál "drágább" áthaladni azon a pixelen):

NÖSZTÉP kategória	Költség érték
1110	100
1210	1
1220	5
1310	5
1410	15
1420	10
2100	15
3120	10
3200	10
3400	10
3500	15
4402	20
4404	20
4600	20
5110	50
5120	40
6100	100
6200	100

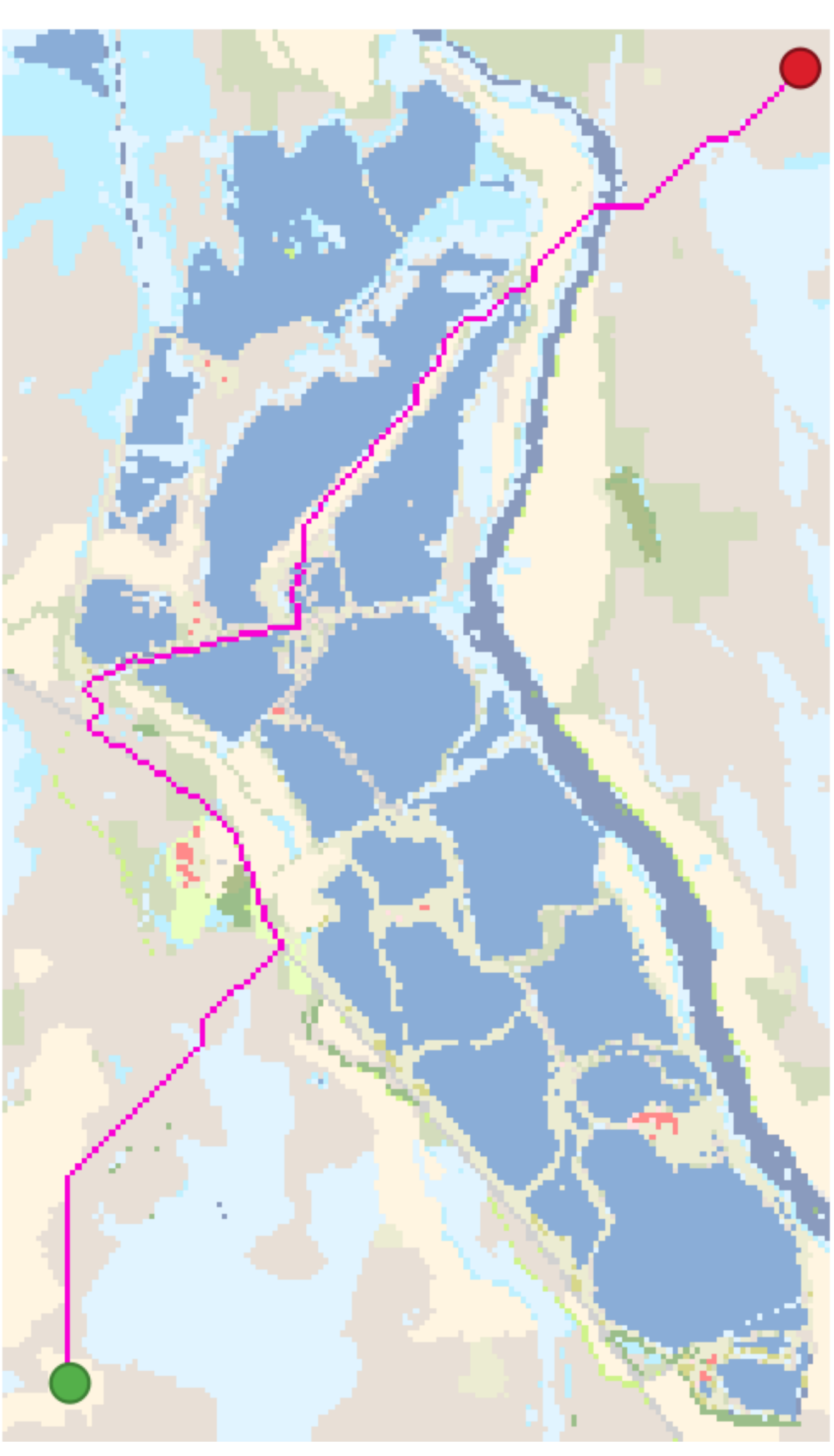
Ezt a költség érték táblázatot felhasználva határozd meg a **legkisebb költségű** (tehát nem légvonalban a legrövidebb!) utat START-tól STOP-ig!

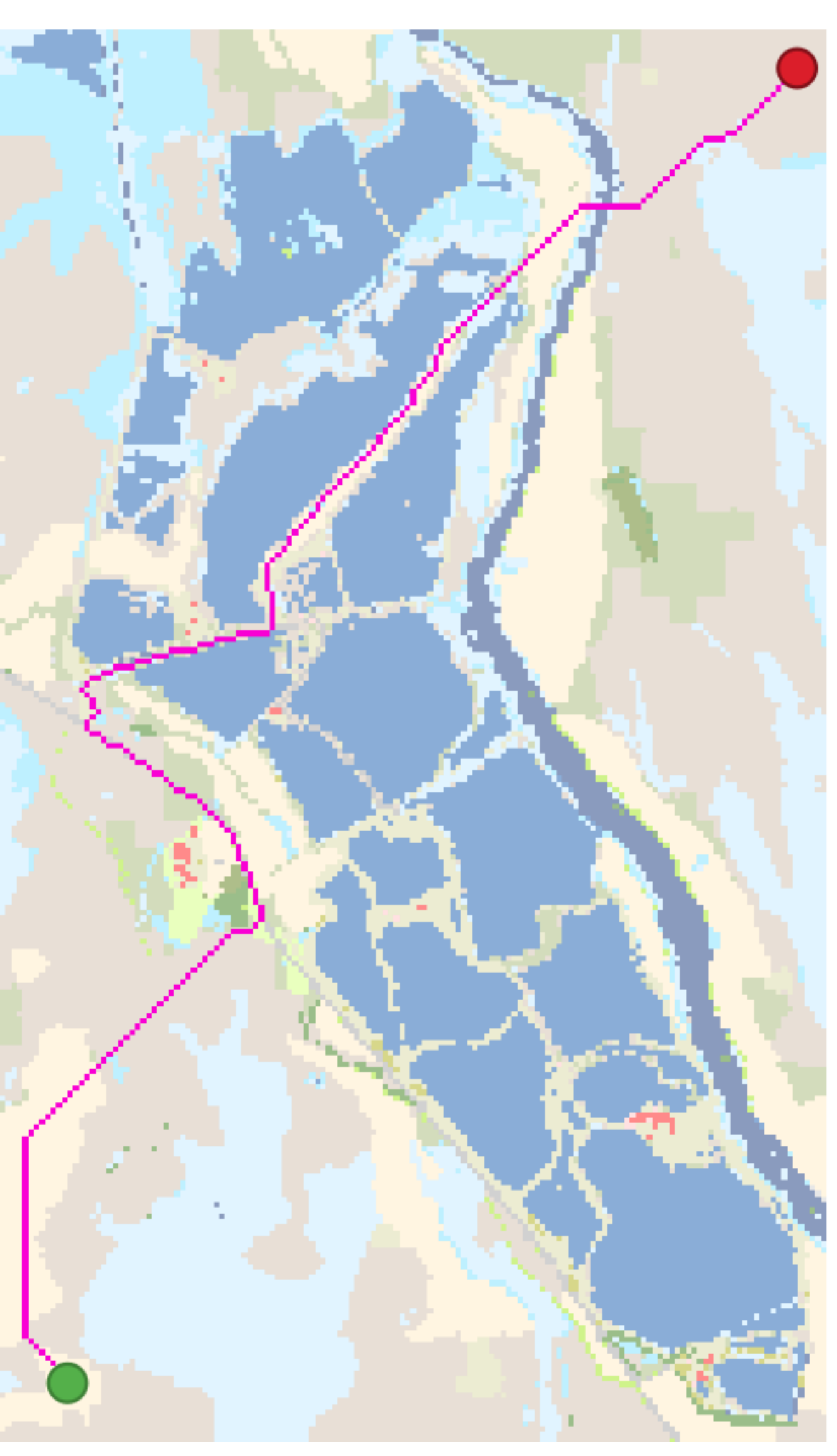
Javasolt szoftverek: *GRASS GIS, QGIS*.

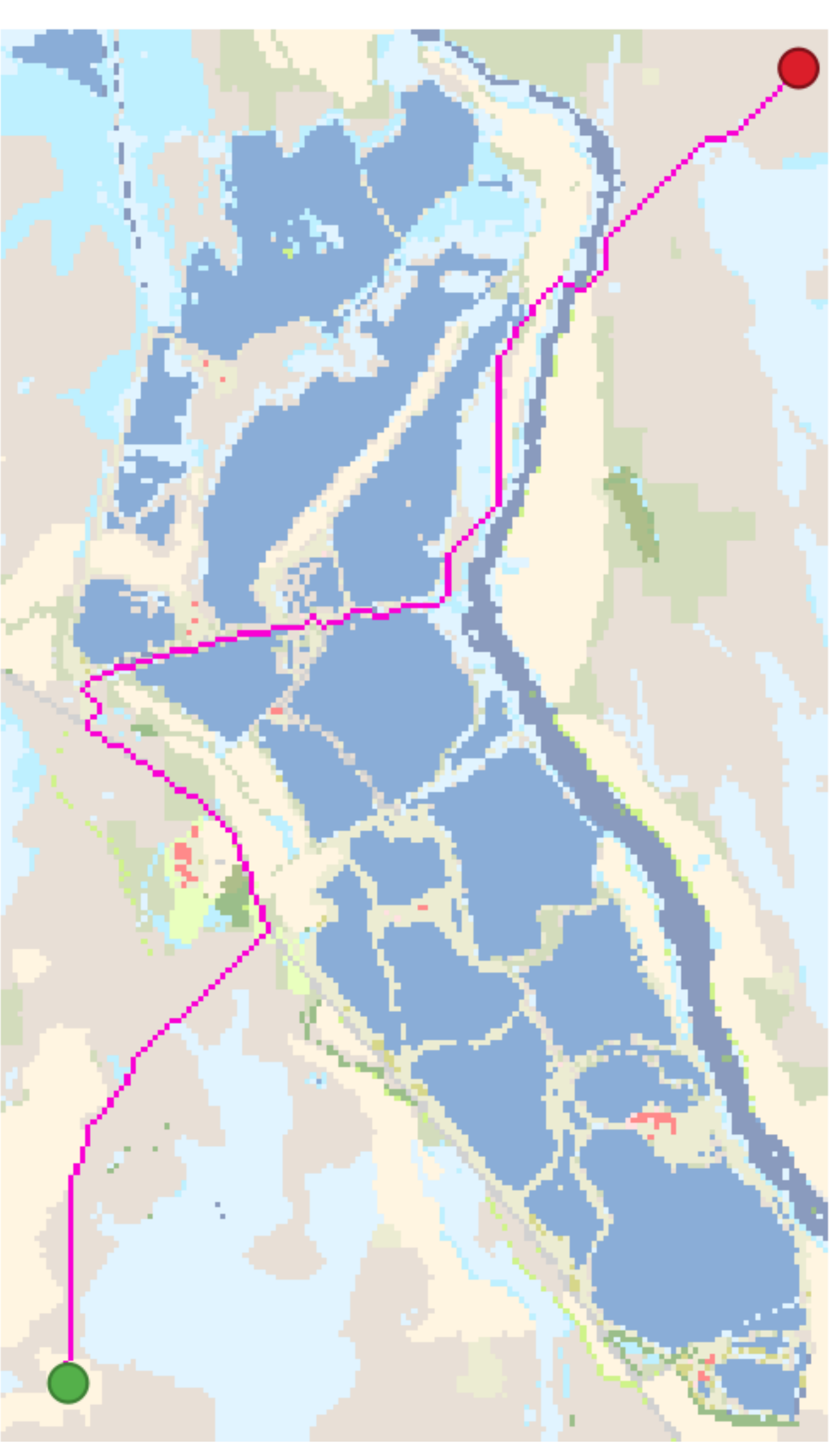
Javasolt GRASS GIS parancsok: *r.reclass, r.cost, r.drain*.

Mellékletekben éred el:

- nosztep.tif: NÖSZTÉP kivágat
- start.shp: kezdőpont
- stop.shp: végpont
- reclass.txt: újraosztályozáshoz felhasználható költség táblázat
- nosztep_style.qml: QGIS stílus fájl a nosztep.tif fájlhoz (csak megjelenítéshez hasznos, a feladat megoldásához nem szükséges)
- grassdata könyvtár: előre konfigurált GRASS GIS induló könyvtár, importálva az alaptérkép raszterrel (nosztep.tif), illetve a start és stop vektorokkal (start.shp, stop.shp).

○


○


●


Magyarázat a megoldáshoz

A feladat egy klasszikus "Least cost path" analízis, melyet most GRASS GIS-ben oldunk meg.

GRASS GIS-be belépés és a bemenő adatok importálása után (az előre konfigurált grassdata könyvtárt használva az importálás nem szükséges) a következő GRASS parancsokra van szükség:

1. lépés: raszter újraosztályozás r.reclass paranccsal, rule fájl segítségével

```
# Raszter újraosztályozás rule fájl segítségével.
# A reclass.txt teljes elérési útját kell megadni
r.reclass input=nosztep output=reclass rules=C:\test\reclass.txt
```

2. lépés: cost surface előállítása r.cost segítségével

```
r.cost input=reclass output=cost outdir=directions start_points=s
```

3. lépés: legkisebb költségű út számítása a cost surface raszterből r.drain parancs segítségével

```
r.drain input=cost direction=directions output=drain start_points
```

A "drain" raszter réteg tartalmazza a végeredményt.

2. feladat 0 / 3 pont

Tavak területe

Határozd meg a rendelkezésre álló OpenStreetMap adatbázisban az Eger belterületén (belterület: "residential") található vízfelületek összterületét! Vízfelületnek minősül minden természetes és nem természetes eredetű, összefüggő területet alkotó állóvíz. Az eredményt hektárban, 3 tizedesjegyre kerekítve (tizedesnek pontot használva) kérjük megadni.

Az adatbázist így tudod elérni:

Terminál ablakba írd be a következő parancsot:

```
docker run -p 5432:5432 --rm -it -e POSTGRES_PASSWORD=postgres fegyi001/
```

Ennek hatására letöltésre, majd elindításra, illetve OpenStreetMap adatokkal feltöltésre kerül egy PostGIS adatbázis az 5432-es porton (amennyiben már használod ezt a portot a gépeden, úgy az "5432:5432" első tagját írd át egy szabad portra, pl. "5433:5432"). Ez a művelet akár 1-2 percet is igénybe vehet a számítógéped kapacitástól függően.

Miután az adatbázis sikeresen felépült (a terminálban az utolsó sor ilyen: "*LOG: database system is ready to accept connections*", adatbázis kezelővel (DBeaver, pgAdmin stb.) tudsz kapcsolódni a "postgres" adatbázishoz:

- `host: localhost`
- `port: 5432`
- `database: postgres`
- `user: postgres`
- `password: postgres`

A "public" sémában találod az OpenStreetMap tematikus rétegeit.

Szükséges szoftver: *Docker, DBeaver/pgAdmin*

A megoldások:
0.237
0.247

Magyarázat a megoldáshoz

PostGIS műveletekkel, ill. SQL where feltételekkel megkapható a kért érték. Fontos, hogy az adatok WGS84-ben vannak tárolva (EPSG:4326), vagyis a végső számításnál kell végezni egy EOV transzformációt. Elfogadjuk az UTM-34 (EPSG:32634) transzformációt is, mindkettő ugyanazt az eredményt adja 3 tizedesjegy pontosság esetén. Hibásnak számít az EPSG:3857 vagy az EPSG:32633 transzformáció (előbbi közismerten csak megjelenítésre való vetületi rendszer, utóbbi az UTM-33, aminek határán kívül esik Eger).

```
select
    sum(st_area(st_transform(w.geom, 23700)))/ 10000 as area_hect
from
    water_a w
join landuse l on
    ST_Within(w.geom,
    l.geom)
join places p on
    ST_Within(p.geom,
    l.geom)
where
    l.fclass = 'residential'
    and p.name = 'Eger';
```

Vissza a kategóriáimhoz