МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

Лабораторна робота № 3

з курсу «Оброблення надвеликих масивів даних»

Тема: «Rendering map data»

Виконав:

студент I курсу

групи ДА-71мн

Колінько Анжела

Київ – 2017

Варіант 2

1. Парсинг файлів

В роботі використовувались записи RT1 (поля

Field BV Fmt Type Beg End Len Description

TLID No R N 6 15 10 TIGER/Line ID, Permanent Record Number

FRLONG No R N 191 200 10 Start Longitude

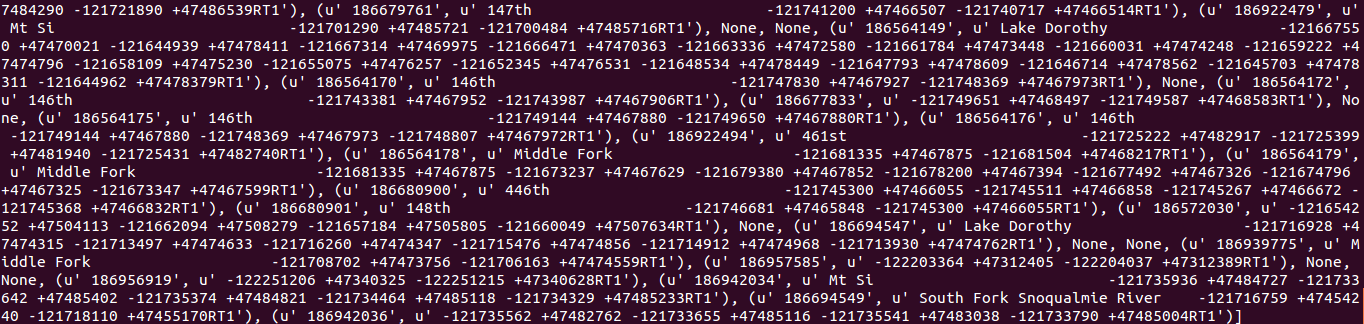
FRLAT No R N 201 209 9 Start Latitude

TOLONG No R N 210 219 10 End Longitude

TOLAT No R N 220 228 9 End Latitude)

та RT2 (всі поля крім Version Number та Record Sequence Number). З датасету було вибрано перші 200 записів з кожного з файлів (RT1 RT2), отримані файли були поміщені в папку input. Далі запіскалися скрипти обробки (див. додаток 1).

Результати обробки файлів записувались до файлу outp\_f.txt. (див. додаток 2)

 – фрагмент виводу скрипту.

2. Нанесення даних на карту.

Дані буреться з файлу outp\_f.txt (користувач повинен вказати на цей файл, адже javascript в цілях безпеки не дозвляє вказувати default файл для загрузки) і відмальвуються дані на карті (було використано google maps api); якщо розміри об’єкту більші за порогові для даного рівня наближення, об’єкт відмальовується (

var tmp\_sq=Math.abs(base\_crds[0]-base\_crds[2])\*Math.abs(base\_crds[1]-base\_crds[3])

var elem\_sq=Math.abs(Math.log(tmp\_sq)/Math.log(2))/200;

var thrh=1/(map.getZoom());

if(elem\_sq>=thrh||map.getZoom()==21){//21 -- max zoom;

…

}). При зміні рівня наближення об’єкти перемальовуються (

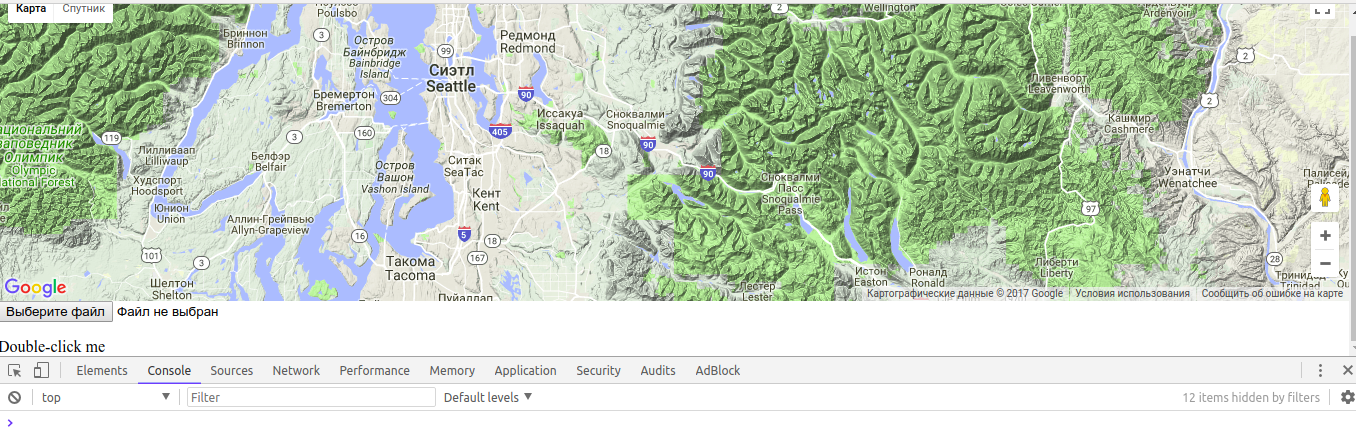
map.addListener('zoom\_changed', function() {

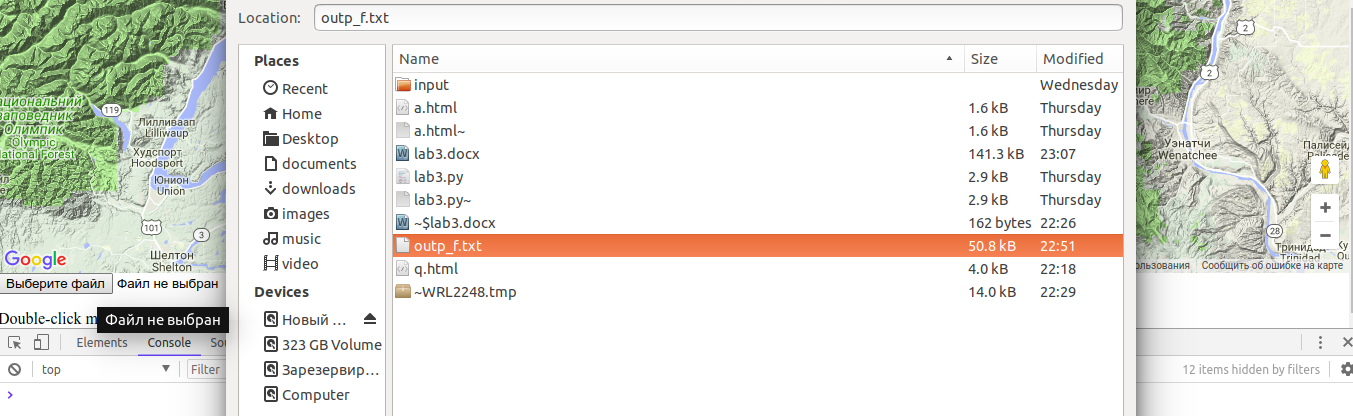
console.log('zoom=' + map.getZoom());

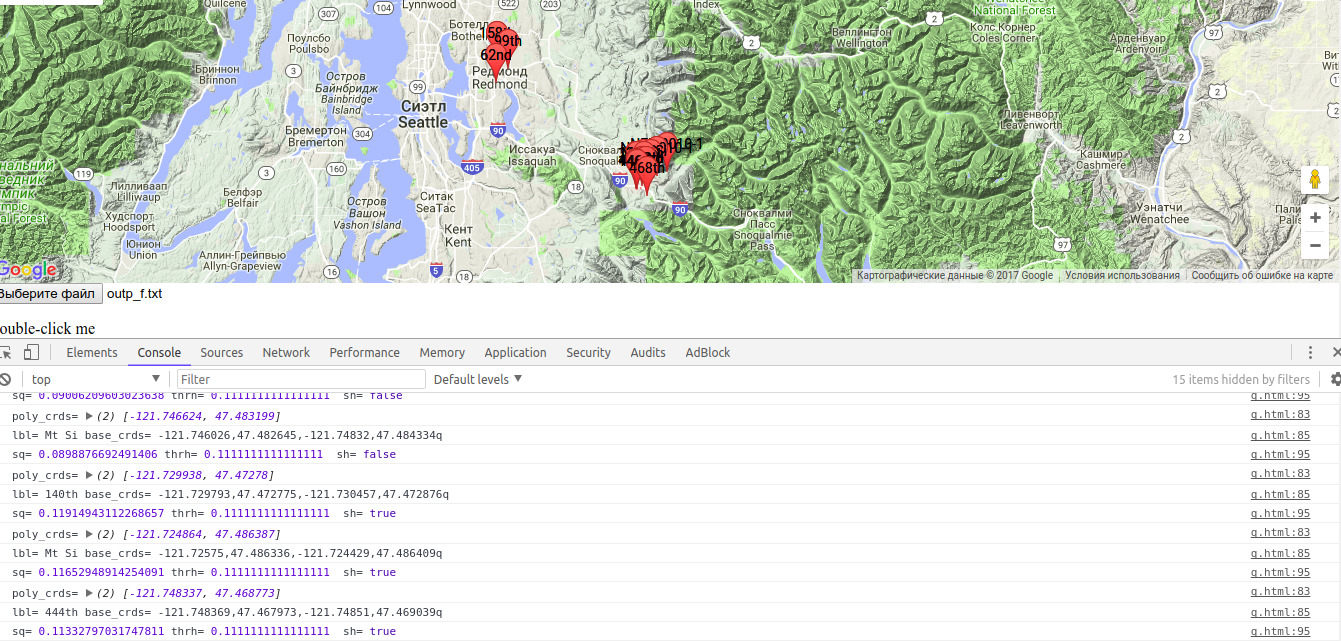
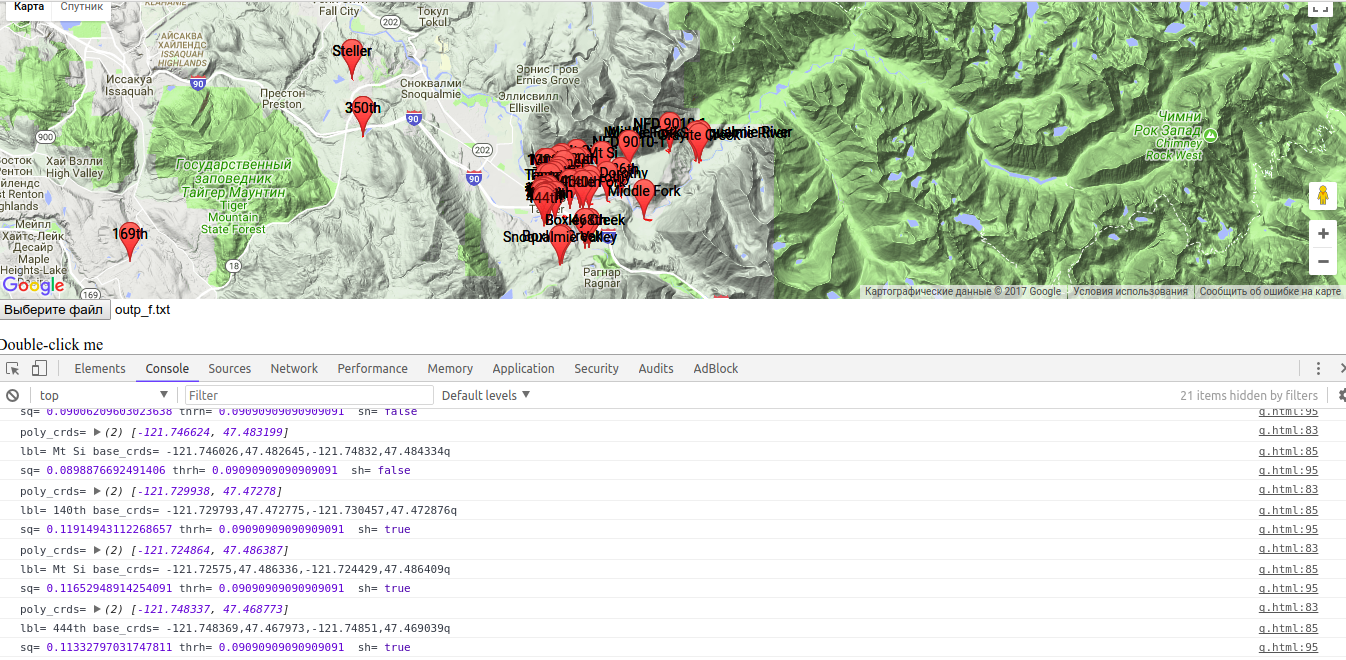
show\_contents();

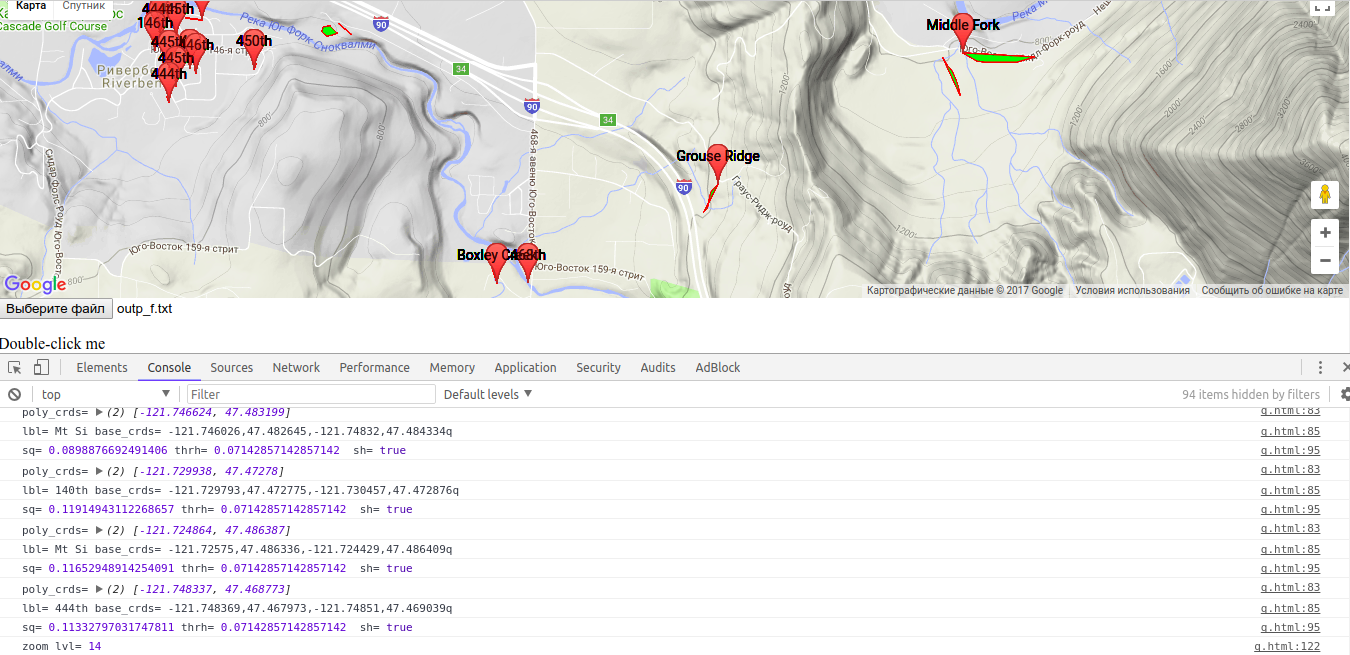
});)

(див. додаток 3)







Додаток 1. lab3.py

def get\_lines\_ff(pair):

f,text=pair;

rt=f[len(f)-3:]; cnt=0

print 'filename=%s; rt==%s' % (f, rt)

n\_text='';

for lines in text.splitlines():

if rt=='RT1':

n\_text+=lines[5:15]+' '#get TIGER/Line ID;

if lines[19:49].replace(' ',''):

if cnt<10:

print 'not empty name: %s %s' % (lines[5:15], lines[19:49])

cnt+=1

n\_text+=lines[19:49]+' '#get Feature Name;

n\_text+=lines[190:200]+' '#get Start Longitude;

n\_text+=lines[200:209]+' '#get Start Latitude;

n\_text+=lines[209:219]+' '#get End Longitude;

n\_text+=lines[219:228]+' '#get End Latitude;

else:

n\_text+=lines[5:15]+' '#get TIGER/Line ID;

#n\_text+=lines[15:18]+' '#get Record Sequence Number;

i=18; lgtd=True

while i<209:

if lgtd:

if lines[i+1:i+10].replace('0',''):

if cnt<10:

print 'not empty point lgtd: %s %s' % (lines[6:15], lines[i:i+10])

cnt+=1

n\_text+=lines[i:i+10]+' '#get Point Longitude;

i+=10;

else:

if lines[i+1:i+9].replace('0',''):

if cnt<10:

print 'not empty point lgtd: %s %s' % (lines[6:15], lines[i:i+9])

cnt+=1

n\_text+=lines[i:i+9]+' '#get Point Latitude;

i+=9;

lgtd=not(lgtd)

n\_text=n\_text[:len(n\_text)-1]#deleting ending ' ' symbol;

n\_text+=rt+'@';

cnt=0; print '\n\n'

return [n\_text];

def group\_f(x,y):

if x==None or y==None:

print 'none'

return

ret=''

if x[len(x)-3:len(x)]=='RT2':

if y[len(y)-3:len(y)]=='RT2':

x=x[:len(x)-3]#clear 'RT2' string;

ret=x+y#+' x=rt2; y=rt2'

elif y[len(y)-3:len(y)]=='RT1':

y=y[:len(y)-3]

x=x[:len(x)-3]+'RT1'#+' x=rt2, y=rt1'

ret=y+x#append rt2 in the end of rt1

else:

print '\nx=RT2; y=?\n'

elif x[len(x)-3:len(x)]=='RT1':

if y[len(y)-3:len(y)]=='RT2':

x=x[:len(x)-3]

y=y[:len(y)-3]+'RT1'

ret=x+y#+' x=rt1, y=rt2'

elif y[len(y)-3:len(y)]=='RT1':

print '\n\nwarning\n\n'

else:

print '\nx=RT1; y=?\n'

else:

print '\nx=?; y=?\n'

print(x)

print(y)

return

#print 'ret=',x,y,ret,'\n'

return ret

def ptf(inp\_tuple):

lid=inp\_tuple[0]

l\_data=inp\_tuple[1]

is\_str=False

if l\_data==None:

print 'l\_data==None'

return

print 'inp'+l\_data+'\n'

if l\_data[len(l\_data)-3:len(l\_data)]=='RT2':

print 'ptf: warning l\_data=%s' % l\_data

return

print 'l\_data(:11)=%s' % l\_data[:11]

try:

float(l\_data[:11])

except ValueError:

is\_str=True

with open("../lab3/outp\_f.txt","ab") as fout:

if is\_str:

fout.write(l\_data[1:len(l\_data)]+'\n')

else:

i=0; spl=''

while i<30:

spl+=' '

i+=1

fout.write(spl+l\_data[0:len(l\_data)]+'\n')

fout.close()

return inp\_tuple

rdd = sc.wholeTextFiles("../lab3/input").flatMap(get\_lines\_ff).flatMap(lambda a: a.split('@')).flatMap(lambda elem: [(elem[:10],elem[10:])]).reduceByKey(group\_f).map(ptf); rdd.collect()

Додаток 2. Фрагмент файлу outp\_f.txt

Mt Si -121721695 +47486549 -121720137 +47486631RT1

-121633478 +47500974 -121639899 +47495335 -121635422 +47499829 -121636290 +47498746 -121637061 +47497897 -121638322 +47496777 -121639654 +47495706RT1

445th -121746681 +47465848 -121747424 +47466739 -121746951 +47465940 -121747391 +47466556RT1

444th -121747354 +47464888 -121749109 +47466555 -121747556 +47465002 -121747760 +47465322RT1

445th -121746444 +47465071 -121746681 +47465848RT1

130th -121700515 +47482617 -121696640 +47481641 -121700454 +47482225 -121700186 +47481923 -121699411 +47481701 -121697653 +47481983 -121696878 +47482003RT1

Mt Si -121746026 +47482645 -121748320 +47484334 -121746624 +47483199RT1

140th -121729793 +47472775 -121730457 +47472876 -121729938 +47472780RT1

Mt Si -121725750 +47486336 -121724429 +47486409 -121724864 +47486387RT1

444th -121748369 +47467973 -121748510 +47469039 -121748337 +47468773RT1

Grouse Ridge Access -121700164 +47460262 -121711703 +47466430 -121702083 +47462772 -121703872 +47463577 -121708434 +47465140RT1

-121633478 +47500974 -121623725 +47510611 -121633276 +47501272 -121632432 +47501957 -121631656 +47502437 -121631420 +47502734 -121631216 +47503716 -121630913 +47504059 -121630339 +47504355 -121628551 +47504857 -121627707 +47505611 -121626213 +47507645 -121624826 +47509589 -121624523 +47509840 -121624153 +47510320RT1

145th -121743348 +47468638 -121744325 +47468729RT1

447th -121744325 +47468729 -121744427 +47469620RT1

445th -121746854 +47468751 -121747664 +47469824 -121746988 +47468773 -121747022 +47468933RT1

446th -121745734 +47468741 -121745700 +47468939RT1

145th -121745734 +47468741 -121746854 +47468751RT1

-121748934 +47468925 -121748510 +47469039RT1

-121663124 +47500254 -121662094 +47508279 -121663223 +47500459 -121663054 +47501533 -121662750 +47501921 -121662615 +47502584 -121662278 +47502812 -121661198 +47503132 -121659275 +47504594 -121658398 +47505622 -121658432 +47505896 -121658634 +47506011 -121660084 +47505531 -121661569 +47504618 -121661872 +47504572 -121662209 +47504686 -121662209 +47504892 -121660927 +47506331 -121660792 +47506628 -121660825 +47507200 -121661331 +47507634 -121662140 +47508136RT1

146th -121747830 +47467927 -121746785 +47467974RT1

Додаток 3. q.html

<!DOCTYPE html>

<html>

<head>

<style>

/\* Always set the map height explicitly to define the size of the div

\* element that contains the map. \*/

#map {

height: 90%;

}

/\* Optional: Makes the sample page fill the window. \*/

html, body {

height: 100%;

margin: 0;

padding: 0;

}

</style>

</head>

<body>

<div id="map"></div>

<input type="file" id="fileinput" />

<script>

var map; var zoom\_lvl=9;

function initMap() {

map = new google.maps.Map(document.getElementById('map'), {

zoom: zoom\_lvl,

center: new google.maps.LatLng(+47.486549,-121.721695 ),

mapTypeId: 'terrain'

});

}

var contents;

function readSingleFile(evt) {

//Retrieve the first (and only!) File from the FileList object

var f = evt.target.files[0];

if (f) {

var r = new FileReader();

r.onload = function(e) {

contents = e.target.result;

console.log(typeof(contents))

show\_contents();

/\*alert( "Got the file.n"

+"name: " + f.name + "n"

+"type: " + f.type + "n"

+"size: " + f.size + " bytesn"

+ "starts with: " + contents.substr(1, contents.indexOf("n"))

);\*/

}

r.readAsText(f);

} else {

alert("Failed to load file");

}

map.addListener('zoom\_changed', function() {

console.log('zoom=' + map.getZoom());

show\_contents();

});

}

document.getElementById('fileinput').addEventListener('change', readSingleFile, false);

</script>

<script async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDB3OTvr8oDKA0-2hhK2noATefSQ\_HO4AQ&callback=initMap">

</script>

<p ondblclick="show\_zoom()">Double-click me</p>

<script>

var fl\_arr;

function show\_contents(){

console.log('hi'+contents.substr(0,10)+contents.split('\n').length+contents.split('\n')[0]);

fl\_arr=contents.split('\n');

for(i=0;i<fl\_arr.length; i++){

var lbl=fl\_arr[i].substring(0,30);

var base\_crds=new Array; var poly\_crds=new Array;

for(j=0;j<fl\_arr[i].substring(31,72).split(' ').length;j++){

var pflt=(parseFloat(fl\_arr[i].substring(31,72).split(' ')[j]));

pflt/=1000000;//Longitude; 10

//else pflt/=100000;//Latitude; 9

base\_crds.push(pflt);

}

if((fl\_arr[i].length-3)>73){

for(j=0;j<fl\_arr[i].substring(73,fl\_arr[i].length-3).split(' ').length;j++){

var pflt=(parseFloat(fl\_arr[i].substring(73,fl\_arr[i].length-3).split(' ')[j]));

pflt/=1000000;

poly\_crds.push(pflt);

}

if(i<10) console.log('poly\_crds=', poly\_crds);

}

if(i<10) console.log('lbl=',lbl.trim(),'base\_crds=',base\_crds+'q');

var crds\_ll=new Array;

for(j=0;j<poly\_crds.length;j+=2){

var ll=new google.maps.LatLng(poly\_crds[j+1], poly\_crds[j])

crds\_ll.push(ll);

//if(i<10) console.log("ll=",ll);

}

var tmp\_sq=Math.abs(base\_crds[0]-base\_crds[2])\*Math.abs(base\_crds[1]-base\_crds[3])

var elem\_sq=Math.abs(Math.log(tmp\_sq)/Math.log(2))/200;

var thrh=1/(map.getZoom());

if(i<10) console.log('sq=',elem\_sq,'thrh=',thrh,' sh=',elem\_sq>=thrh);

if(elem\_sq>=thrh||map.getZoom()==21){//21 -- max zoom;

var rect = new google.maps.Polygon({

paths: crds\_ll,

strokeColor: '#FF0000',

strokeOpacity: 0.7,

strokeWeight: 1,

fillColor: '#00FF00',

fillOpacity: 0.35

});

rect.setMap(map);

if(lbl.trim()) addMarker(crds\_ll[0], map,lbl.trim());

}

function addMarker(location, map,lbl) {

// Add the marker at the clicked location, and add the next-available label

// from the array of alphabetical characters.

var marker = new google.maps.Marker({

position: location,

label: lbl,//labels[labelIndex++ % labels.length],

map: map

});

}

}

}

</script>

<script>

function show\_zoom(){

console.log("zoom lvl=", map.getZoom());

}

</script>

</body>

</html>