

## Test Task: Automatic Land Plot Development Algorithm

### Task Description

You need to develop an algorithm for automatically distributing areas within a land plot based on given parameters. The input consists of the land plot coordinates in GeoJSON format and key urban planning constraints. The output should be a GeoJSON file with information about the placement of objects and an image of the development plan.

### Input Data

The program must accept the following parameters:

- **Plot boundaries** – provided in GeoJSON format (FeatureCollection with Polygon).
- **Building density constraints** – percentage of the plot area allowed for construction.
- **Minimum distance between objects** – for example, 10 meters between buildings.
- **Additional constraints** – presence of roads, forests, bodies of water where construction is prohibited (also provided in GeoJSON format).

### Algorithm Objectives

- Divide the plot into logical zones (e.g., residential development, parks, roads).
- Place objects (buildings, roads, green areas) considering the given constraints, excluding construction in prohibited zones (roads, forests, bodies of water).
- Generate a GeoJSON file with the coordinates of the placed objects.
- Generate an image of the plot development plan.

### Output Data

- A **GeoJSON file** containing information about the objects on the plot in the format:

```
{  
  "type": "FeatureCollection",  
  "features": [  
    {  
      "type": "Feature", "properties": {"type": "residential"}, "geometry": {"type": "Polygon",  
        "coordinates": [[...]]},
```

```
    {"type": "Feature", "properties": {"type": "commercial"}, "geometry": {"type": "Polygon",  
"coordinates": [[...]]}},
```

```
    {"type": "Feature", "properties": {"type": "road"}, "geometry": {"type": "LineString",  
"coordinates": [[...]]}},
```

```
    {"type": "Feature", "properties": {"type": "park"}, "geometry": {"type": "Polygon",  
"coordinates": [[...]]}}
```

```
]
```

```
}
```

- A **.png or .jpg image** showing a graphical representation of the land plot development.

### **Submission Format**

- The solution must be presented in a Jupyter Notebook (.ipynb).
- Python is allowed, along with libraries such as matplotlib, shapely, geopandas, json, and others.
- The code should be clean, well-commented, and include explanations of the logic.

### **Evaluation Criteria**

- Correct functioning of the algorithm (respecting constraints, no object overlaps).
- Code structure and readability.
- Clarity of the result visualization.

### **Bonus Points**

- Generation of multiple development variants on each run.
- Ability to handle various plot shapes (square, rectangular, arbitrary).
- Consideration of additional factors (e.g., access to public transport).