Laboratorium 2
Julia Skoroszewska, Emilia Gnatiuk

Wersja implementacji: Podstawowa wersja algorytmu Gillespiego, w którym model jest wpisany w algorytm.
Podczas podstawowego algorytmu Gillespiego każda reakcja jest symulowana osobno, losując czas do kolejnego zdarzenia i wybierając, która z reakcji zajdzie na podstawie aktualnej szybkości.

Pliki wejściowe są w formacie .json.

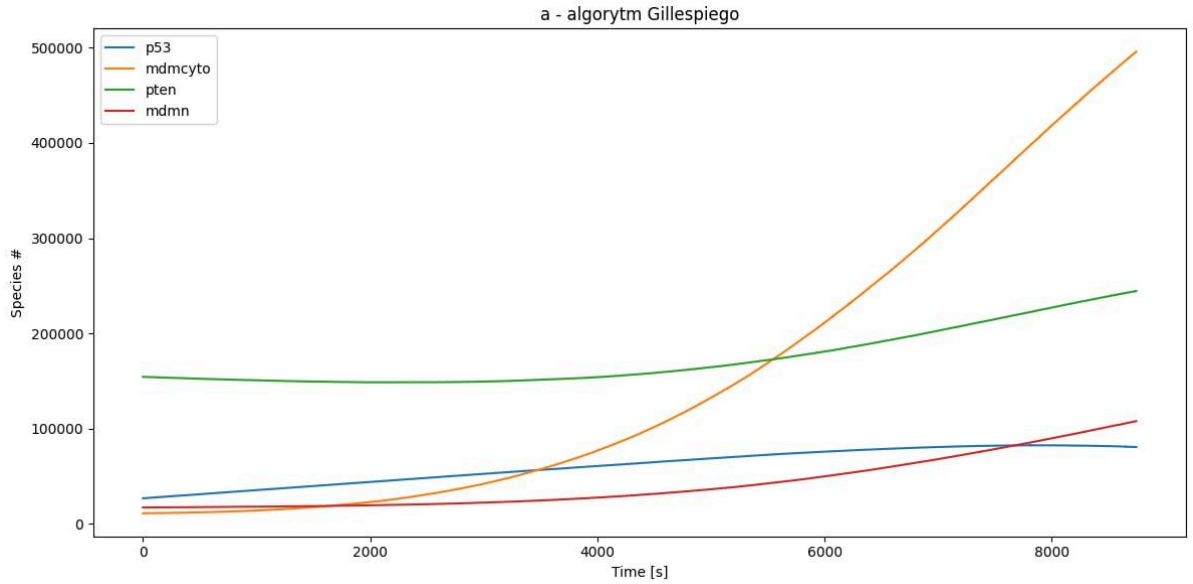
Kod:

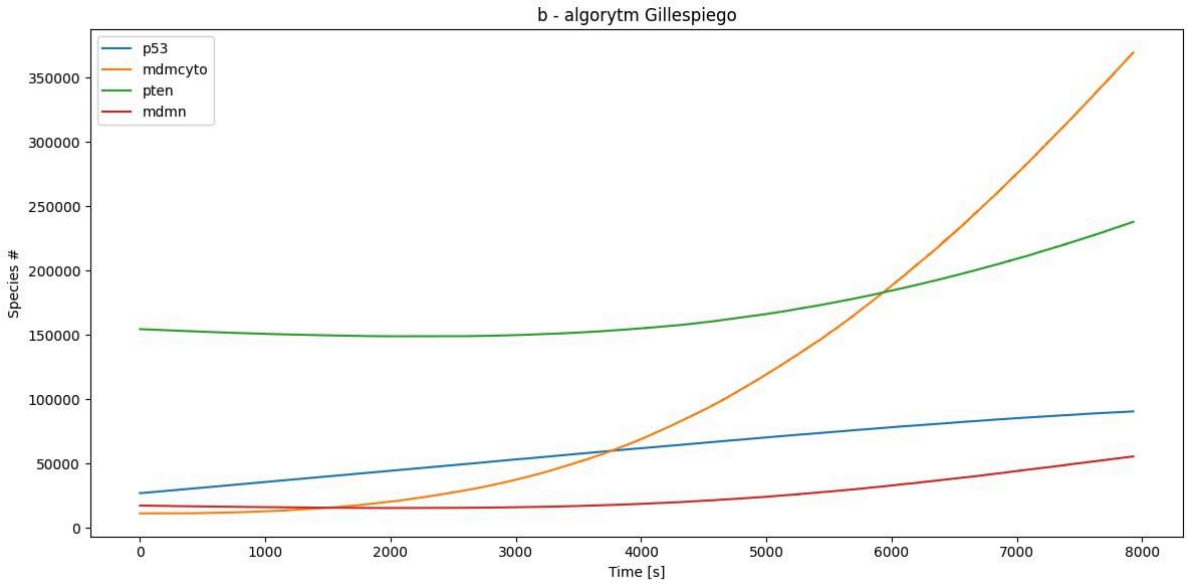
```
import os
import json
import random
import numpy as np
from scipy.stats import poisson

import matplotlib.pyplot as plt

def szybkosc_reakcji(reaction):
    all_vars = reaction["params"] + reaction["species"]
    lambda_str = f"lambda {', '.join(all_vars)}:
{reaction['equation']}"
    return eval(lambda_str), all_vars

def algorytm_Gillespiego(model, max_steps, max_time=None):
    sp_names = list(model['species_init'].keys())
    params = model['parameters']
    sp_idx = {name: i for i, name in enumerate(sp_names)}
    n_species = len(sp_names)
    sp_traj = np.zeros((max_steps + 1, n_species))
    for i, name in enumerate(sp_names):
        sp_traj[0, i] = model['species_init'][name]
    t_traj = np.zeros((max_steps + 1))
    rxn_list = []
    for rxn in model['reactions'].values():
        rate_fn, var_names = szybkosc_reakcji(rxn)
        rxn_list.append({'rate_fn': rate_fn, 'vars': var_names,
'effects': rxn['effects']})
    steps_done = 0
    for step in range(max_steps):
        now = t_traj[step]
        if max_time is not None and now >= max_time:
            break
        curr_sp = sp_traj[step]
        rates = []
        for rxn in rxn_list:
```

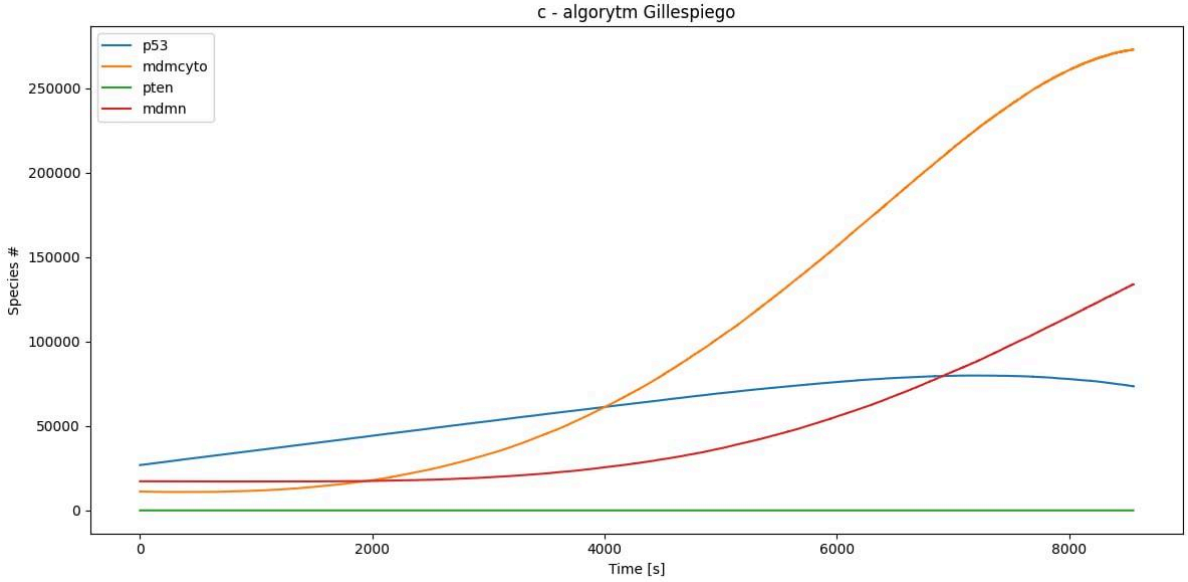
```
            vals = []
            for v in rxn['vars']:
                if v in params:
                    vals.append(params[v])
                else:
                    vals.append(curr_sp[sp_idx[v]])
            try:
                r = rxn['rate_fn'](*vals)
            except Exception:
                r = 0
            rates.append(max(r, 0))
        total_r = sum(rates)
        if total_r <= 0:
            break
        dt = random.expovariate(total_r)
        t_traj[step + 1] = now + dt
        sp_traj[step + 1] = curr_sp.copy()
        threshold = random.uniform(0, total_r)
        acc = 0
        chosen = -1
        for i, r in enumerate(rates):
            acc += r
            if threshold <= acc:
                chosen = i
                break
        rxn = rxn_list[chosen]
        for sp, eff in rxn['effects'].items():
            idx = sp_idx[sp]
            sp_traj[step + 1][idx] += eff
        steps_done = step + 1
    return t_traj[:steps_done + 1], sp_traj[:steps_done + 1],
sp_names


def main():
    scenarios_dir = os.path.join(os.getcwd(), 'scenariusze')
    plots_dir = os.path.join(os.getcwd(), 'wykresy')
    os.makedirs(plots_dir, exist_ok=True)
    scenarios = {}
    for filename in os.listdir(scenarios_dir):
        if filename.endswith('.json'):
            with open(os.path.join(scenarios_dir, filename), 'r',
encoding='utf-8') as f:
                scenario_name = os.path.splitext(filename)[0]
                scenarios[scenario_name] = json.load(f)
    max_steps = 10000
    max_time = 172800
    for selected, scenario_model in scenarios.items():
```

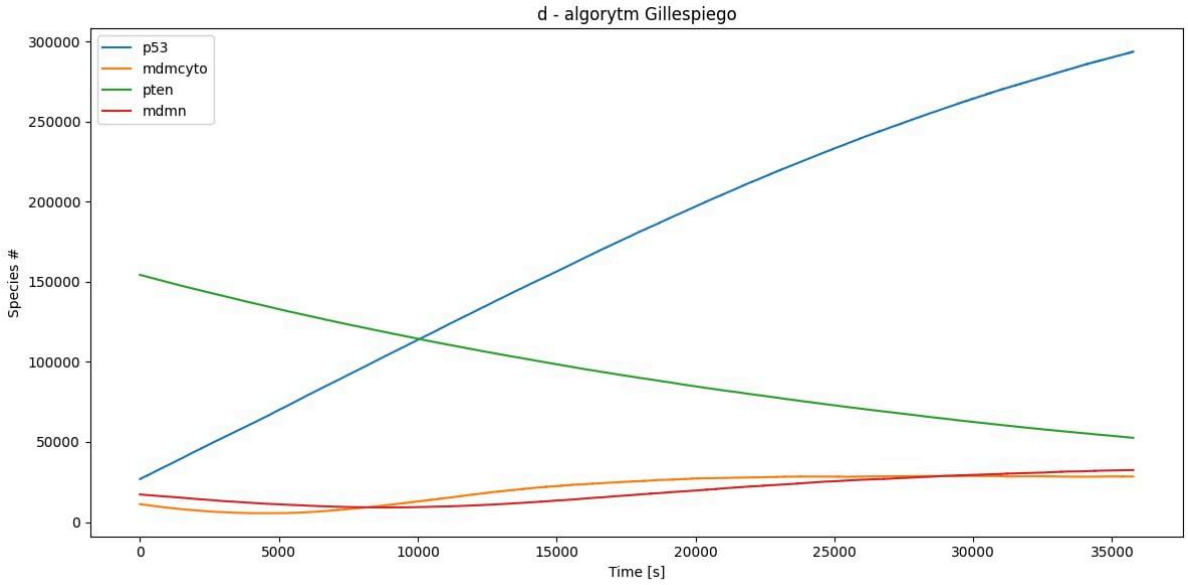
```
        t_direct, sp_direct, sp_names = algorytm_Gillespiego(
            scenario_model, max_steps=1000000, max_time=max_time
        )
        plt.figure(figsize=(12, 6))
        for idx, label in enumerate(sp_names):
            plt.plot(t_direct, sp_direct[:, idx], label=label)
        plt.xlabel('Time [s]')
        plt.ylabel('Species #')
        plt.title(f"{selected} - algorytm Gillespiego")
        plt.legend()
        plt.tight_layout()
        plot_path = os.path.join(plots_dir,
f"{selected}_algorytm_Gillespiego.png")
        plt.savefig(plot_path)
        plt.close()


if __name__ == "__main__":
    main()
```

Wykresy:




b - algorytm Gillespiego
p53
mdmcyto
pten
mdmn
Species #
Time [s]
0
50000
100000
150000
200000
250000
300000
350000
0
1000
2000
3000
4000
5000
6000
7000
8000


c - algorytm Gillespiego
p53
mdmcyto
pten
mdmn
Species #
Time [s]
0
50000
100000
150000
200000
250000
0
2000
4000
6000
8000


d - algorytm Gillespiego
p53
mdmcyto
pten
mdmn
Species #
Time [s]
0
50000
100000
150000
200000
250000
300000
0
5000
10000
15000
20000
25000
30000
35000