

# Лабораториска вежба 1 - Маркови процеси на одлучување

## Gym - Python библиотека

Gym е Python библиотека која поддржува развој и споредба на алгоритми за учење со поттикнување. Документацијата на оваа библиотека е достапна на следната [страница](#). Инсталација на библиотеката:



```
pip install gymnasium
```

За компатибилност со примерите од аудиториските вежби користете ја верзијата 0.18.0.





Околина може да се креира на следниот начин:

```
env = gym.make(env_name)
```

каде env\_name е името на околината. Библиотеката gym обезбедува голем број на [околин](#) кои може да се користат. Пример за околин:

-  FrozenLake-v0
-  Taxi-v3

Некои атрибути на околината кои може да се користат:

-  Action\_space - валидни акции
-  Observation\_space - валидни состојби
-  Reward\_range - ранг на наградата која агентот ја добива
-  Env - дополнителни информации за околината

Околината се ресетира со повик на функцијата reset():

```
env.reset()
```

Оваа функција ја враќа почетната состојба.

Со повик на функцијата render() може да се визуелизира тековната состојба:

```
env.render()
```

За движење низ околината се користи функцијата step:

```
state, reward, done, info = env.step(action)
```

каде action е акцијата која се презема. Оваа функција како резултат враќа 4 вредности:

- State - објект кој ја претставува следната состојба на околината
- Reward - награда која се добива по преземање на акцијата action
- Done - информација дали епизодата е завршена (ако е завршена, околината треба да се ресетира)
- Info - дополнителни информации специфични за конкретната околина

## Задачи

### Задача 1 (25 поени)

За околината „Taxi-v3“ одредете ја најдобрата политика со **value iteration** со следните вредности за `discount_factor`: 0.5, 0.7 и 0.9. Може да ја користите имплементацијата достапна во скриптата **mdp.py**. За секоја состојба одредете ја најдобрата акција која треба да се преземе. Потоа визуелизирајте го движењето на агентот низ оваа околина.

Тестирајте ја добиената политика во 50 и 100 итерации. Колкав е просечниот број на чекори потребни за стигнување до целта? Колкава е просечната награда?

### Задача 2 (25 поени)

За околината „Taxi-v3“ одредете ја најдобрата политика со **policy iteration** со следните вредности за `discount_factor`: 0.5, 0.7 и 0.9. Може да ја користите имплементацијата достапна во скриптата **mdp.py**. Дали се добива истата политика како и во претходната задача? За секоја состојба одредете ја најдобрата акција која треба да се преземе. Потоа визуелизирајте го движењето на агентот низ оваа околина.

Тестирајте ја добиената политика во 50 и 100 итерации. Колкав е просечниот број на чекори потребни за стигнување до целта? Колкав е просечниот број на чекори потребни за стигнување до целта споредено со бројот на чекори во претходната задача? Колкава е просечната награда? Колкава е просечната награда споредено со просечната награда во претходната задача?