**Mohamad Ali Ghaddar**

**2022655378**

**TP 1: Simple Model MDP**

**Question 1:**

Run:

----------------------------

Simple MDP

----------------------------

----------------------------

START MDP model

----------------------------

STATES S :

['s0', 's1', 's2', 's3']

ACTIONS A :

['right', 'down', 'left', 'up']

TRANSITION FUNCTION :

start state = s0, action = right, next\_states = {'s0': 1.0}

start state = s0, action = down, next\_states = {'s1': 1.0}

start state = s0, action = left, next\_states = {'s0': 1.0}

start state = s0, action = up, next\_states = {'s0': 1.0}

start state = s1, action = right, next\_states = {'s2': 1.0}

start state = s1, action = down, next\_states = {'s1': 1.0}

start state = s1, action = left, next\_states = {'s1': 1.0}

start state = s1, action = up, next\_states = {'s0': 1.0}

start state = s2, action = right, next\_states = {'s2': 1.0}

start state = s2, action = down, next\_states = {'s2': 1.0}

start state = s2, action = left, next\_states = {'s1': 1.0}

start state = s2, action = up, next\_states = {'s3': 1.0}

start state = s3, action = right, next\_states = {'s3': 1.0}

start state = s3, action = down, next\_states = {'s3': 1.0}

start state = s3, action = left, next\_states = {'s3': 1.0}

start state = s3, action = up, next\_states = {'s3': 1.0}

REWARD FUNCTION :

start state = s0, action = right, reward = -1

start state = s0, action = down, reward = -1

start state = s0, action = left, reward = -1

start state = s0, action = up, reward = -1

start state = s1, action = right, reward = -1

start state = s1, action = down, reward = -1

start state = s1, action = left, reward = -1

start state = s1, action = up, reward = -1

start state = s2, action = right, reward = -1

start state = s2, action = down, reward = -1

start state = s2, action = left, reward = -1

start state = s2, action = up, reward = 10

start state = s3, action = right, reward = 0

start state = s3, action = down, reward = 0

start state = s3, action = left, reward = 0

start state = s3, action = up, reward = 0

----------------------------

END MDP model

----------------------------

----------------------------

ITERATIONS OF MDP VALUE ITERATION

----------------------------

Iteration 1 values: {'s0': -1.0, 's1': -1.0, 's2': 10.0, 's3': 0.0}

Iteration 2 values: {'s0': -1.99, 's1': 8.9, 's2': 10.0, 's3': 0.0}

Iteration 3 values: {'s0': 7.811, 's1': 8.9, 's2': 10.0, 's3': 0.0}

Iteration 4 values: {'s0': 7.811, 's1': 8.9, 's2': 10.0, 's3': 0.0}

----------------------------

OPTIMAL POLICY À PARTIR DE S0

----------------------------

State: s0, Action: down, Next State: s1, Reward: -1

State: s1, Action: right, Next State: s2, Reward: -1

State: s2, Action: up, Next State: s3, Reward: 10

**Question 2 :**

**2,1 :**

Code attaché dans le dossier.

**Run :**

----------------------------

Simple MDP

----------------------------

----------------------------

START MDP model

----------------------------

STATES S :

['s0', 's1', 's2', 's3']

ACTIONS A :

['right', 'down', 'left', 'up']

TRANSITION FUNCTION :

start state = s0, action = right, next\_states = {'s0': 1.0}

start state = s0, action = down, next\_states = {'s1': 0.8, 's2': 0.2}

start state = s0, action = left, next\_states = {'s0': 1.0}

start state = s0, action = up, next\_states = {'s0': 1.0}

start state = s1, action = right, next\_states = {'s2': 0.9, 's3': 0.1}

start state = s1, action = down, next\_states = {'s1': 1.0}

start state = s1, action = left, next\_states = {'s1': 1.0}

start state = s1, action = up, next\_states = {'s0': 0.8, 's3': 0.2}

start state = s2, action = right, next\_states = {'s2': 1.0}

start state = s2, action = down, next\_states = {'s2': 1.0}

start state = s2, action = left, next\_states = {'s1': 0.9, 's0': 0.1}

start state = s2, action = up, next\_states = {'s3': 0.9, 's0': 0.1}

start state = s3, action = right, next\_states = {'s3': 1.0}

start state = s3, action = down, next\_states = {'s3': 1.0}

start state = s3, action = left, next\_states = {'s3': 1.0}

start state = s3, action = up, next\_states = {'s3': 1.0}

REWARD FUNCTION :

start state = s0, action = right, reward = -1

start state = s0, action = down, reward = -1

start state = s0, action = left, reward = -1

start state = s0, action = up, reward = -1

start state = s1, action = right, reward = -1

start state = s1, action = down, reward = -1

start state = s1, action = left, reward = -1

start state = s1, action = up, reward = -1

start state = s2, action = right, reward = -1

start state = s2, action = down, reward = -1

start state = s2, action = left, reward = -1

start state = s2, action = up, reward = 10

start state = s3, action = right, reward = 0

start state = s3, action = down, reward = 0

start state = s3, action = left, reward = 0

start state = s3, action = up, reward = 0

----------------------------

END MDP model

----------------------------

----------------------------

ITERATIONS OF MDP VALUE ITERATION

----------------------------

Iteration 1 values: {'s0': -1.0, 's1': -1.0, 's2': 9.901, 's3': 0.0}

Iteration 2 values: {'s0': 0.16839800000000005, 's1': 7.821791, 's2': 10.016671402, 's3': 0.0}

Iteration 3 values: {'s0': 7.178159409596001, 's1': 7.924854219182, 's2': 10.710637781550004, 's3': 0.0}

Iteration 4 values: {'s0': 7.397190822339045, 's1': 8.543178263361055, 's2': 10.732321891411566, 's3': 0.0}

Iteration 5 values: {'s0': 7.8911969190814455, 's1': 8.562498805247705, 's2': 10.781228494989064, 's3': 0.0}

Iteration 6 values: {'s0': 7.916182295764018, 's1': 8.606074589035256, 's2': 10.783702047280638, 's3': 0.0}

----------------------------

OPTIMAL POLICY À PARTIR DE S0

----------------------------

State: s0, Action: down, Next State: s1, Reward: -1

State: s1, Action: right, Next State: s2, Reward: -1

State: s2, Action: up, Next State: s3, Reward: 10

**2,2 :**

Le nombre d’itérations dans gameGrid4Prob est 6 > 4 itérations de la première fiche gameGrid4, mais l’algorithme se converge vers la solution rapidement vu que c’est un MDP simple.

**Question 3:**

**3,1:**

----------------------------

Simple MDP

----------------------------

----------------------------

START MDP model

----------------------------

STATES S :

['s0', 's1', 's2', 's3']

ACTIONS A :

['right', 'down', 'left', 'up']

TRANSITION FUNCTION :

start state = s0, action = right, next\_states = {'s0': 1.0}

start state = s0, action = down, next\_states = {'s1': 0.8, 's2': 0.2}

start state = s0, action = left, next\_states = {'s0': 1.0}

start state = s0, action = up, next\_states = {'s0': 1.0}

start state = s1, action = right, next\_states = {'s2': 0.9, 's3': 0.1}

start state = s1, action = down, next\_states = {'s1': 1.0}

start state = s1, action = left, next\_states = {'s1': 1.0}

start state = s1, action = up, next\_states = {'s0': 0.8, 's3': 0.2}

start state = s2, action = right, next\_states = {'s2': 1.0}

start state = s2, action = down, next\_states = {'s2': 1.0}

start state = s2, action = left, next\_states = {'s1': 0.9, 's0': 0.1}

start state = s2, action = up, next\_states = {'s3': 0.9, 's0': 0.1}

start state = s3, action = right, next\_states = {'s3': 1.0}

start state = s3, action = down, next\_states = {'s3': 1.0}

start state = s3, action = left, next\_states = {'s3': 1.0}

start state = s3, action = up, next\_states = {'s3': 1.0}

REWARD FUNCTION :

start state = s0, action = right, reward = -1

start state = s0, action = down, reward = -1

start state = s0, action = left, reward = -1

start state = s0, action = up, reward = -1

start state = s1, action = right, reward = -1

start state = s1, action = down, reward = -1

start state = s1, action = left, reward = -1

start state = s1, action = up, reward = -1

start state = s2, action = right, reward = -1

start state = s2, action = down, reward = -1

start state = s2, action = left, reward = -1

start state = s2, action = up, reward = 10

start state = s3, action = right, reward = 0

start state = s3, action = down, reward = 0

start state = s3, action = left, reward = 0

start state = s3, action = up, reward = 0

----------------------------

END MDP model

----------------------------

----------------------------

ITERATIONS OF MDP VALUE ITERATION

----------------------------

Iteration 1 values: {'s0': -1.0, 's1': -1.0, 's2': 9.995, 's3': 0.0}

Iteration 2 values: {'s0': -0.94005, 's1': -0.5502250000000001, 's2': 9.995299750000001, 's3': 0.0}

Iteration 3 values: {'s0': -0.9220560025000001, 's1': -0.5502115112499999, 's2': 9.9953897199875, 's3': 0.0}

----------------------------

OPTIMAL POLICY À PARTIR DE S0

----------------------------

State: s0, Action: down, Next State: s1, Reward: -1

State: s1, Action: right, Next State: s2, Reward: -1

State: s2, Action: up, Next State: s3, Reward: 10

**3,2:**

Le nombre d’itération passe de 6 avec gamma 0.99 à 3 avec gamma = 0.05, avec un gamma plus faible, l'accent est mis sur les récompenses immédiates. Les actions qui offrent des récompenses immédiates plus élevées sont privilégiées, même si elles ne mènent pas nécessairement à des récompenses à long terme optimales.

**Question 4:**

**4,1 :**

Fiche gameGrid6 attachée au dossier.

**4,2 :**

Le nombre d’itérations est à 231 pour le gameGrid6, qui est beaucoup plus grand que 3 itérations pour le chemin à 4 routes dans gameGrid4, c’est-à-dire que même si le nombre de route n’augment pas beaucoup, l’algorithme trouve beaucoup de difficulté et à besoin de plusieurs itérations pour trouvé la solution.