R R R R	30 cash, play slot tate 30 cash, play slot 30 cash, play slot tate 30 cash, play slot tate 10 cash, play slot 10 cash, play slot tate	X, 40 cash) X, 20 cash) Y, 40 cash) Y, 10 cash) X, 20 cash)	= .95 #can = .01 #can = .99 #can = .05 #can	play X, cost play Y, cost play Y, cost play X, cost	s 10 and not w s 20 but can w s 20 and not w s 10 but can w	vin 20 => 40 c vin => 20 c vin 30 => 40 c vin => 10 c vin 20 => 20 c	cash: term cash cash: term cash:
	d: 20 cash, play slot 20 cash, play slot 20 cash, play slot 20 cash, play slot 30 cash, play slot 10 cash, play slot 10 cash, play slot	<pre>X, 10 cash) Y, 30 cash) Y, 0 cash) X, 40 cash) X, 20 cash) Y, 40 cash) Y, 10 cash) X, 20 cash)</pre>	= 0 = 0 = -5 = 10 = 0 = 10 = 0				
In the gym p Ope Read pefore	emainder of the problem ackage to develop our solution AI Gym Setup he set-up instructions for proceeding.	set you will imple lution in Python.					
Insteathe G Winte Comp	en Lake I of using CartPole, we're hub link. is quickly approaching, a stely frozen and if you ma hout falling in. The lake we face is described using a SFFFFFFHFHFHFFFHH	and we have to w ake the wrong ste ve're going to co	orry about nav p you may fall nsider is a squ	rigating frozen lak through. We'll ne	tes. It's only early Need to learn how to	November, so the	lakes haven
• S • G • F • H	pot can have one of four starting point. goal point. frozen spot, where it's say hole in the ice, where it's	afe to walk. s not safe to walk	F		F		F
to trai step y	F H are four possible actions: an agent to discover this ou intended.	s via problem sol [,]	ving. However,	•	•	•	
You re Q2. W Write action termin	ceive a reward of 1 if you alking on the Frozen Lake script that sets up the Frozen during each walk. After eal state, because the ager wironment is static and do	reach the goal, a ke [10 points] rozen Lake enviro each step, rende nt fell into a hole	onment and tal r the current si (stop if it is). Ir	kes 10 walks thro tate, and print ou n your own words	t the reward and w , explain how this e	hether the walk is environment beha	"done", i.e. ves.
Coll Us Requ om g Requ ages Coll Us Requ 3.8/	nstall gym cting gym ng cached gym-0.17. rement already sati (from gym) (1.18.5) cting pyglet<=1.5.0 ng cached pyglet-1. rement already sati ite-packages (from rement already sati	isfied: scipy isfied: numpy 0,>=1.4.0 .5.0-py2.py3- isfied: cloud gym) (1.5.0)	>=1.10.4 ir none-any.wh pickle<1.7.	n /Users/khoan nl (1.0 MB) nl (0,>=1.2.0 in	tran/opt/anaco /Users/khoatr	nda3/lib/pyth an/opt/anacon	on3.8/sit da3/lib/p
Buil Bu Cr 5bb5 St dce6 Succe Inst Succe Note	yglet<=1.5.0,>=1.4. ing wheels for coll lding wheel for gym: ated wheel for gym: 6f4a73af555ea98c77f red in directory: / 4ddec9666c8a5e0a35 ssfully built gym lling collected pac ssfully installed g you may need to re	Lected packag (setup.py) filename=gy fa7f8f698fe3c /Users/khoatr ckages: pygle gym-0.17.3 py	es: gym done m-0.17.3-py 44d584dbe an/Library/ t, gym glet-1.5.0	Caches/pip/wi	heels/84/40/e7		
<pre>impo env # en env. for</pre>	<pre>t gym t numpy as np gym.make('FrozenLa = gym.make('Frozen eset() in range(10): nv.reset() or _ in range(10): env.render() observation, rew print("Reward:", if done: break;</pre>	nLake-v0').un ward, done, i	nfo = env	.step(env.act	ion_space.samp	le()) # <i>take</i>	a random
SFFF FHFH FFFH HFFG Rewa (U) SFFF FHFH FFFH HFFG Rewa	d: 0.0 Done: False						
(D) SFFF HFH FFFH HFFG	d: 0.0 Done: False wn) d: 0.0 Done: True						
(Uj SFFF FHFH FFFH HFFG Rewa	<pre>d: 0.0 Done: False) d: 0.0 Done: False ft)</pre>						
SFFF FHFH HFFG Rewa L SFFF FHFH FFFH	d: 0.0 Done: False ft) d: 0.0 Done: False						
FFFF FHFH FFFG Rewa (R SFFF FHFH FFFH HFFG Rewa	d: 0.0 Done: False ght) d: 0.0 Done: True						
(U) FFFF FHFH FFFH HFFG Rewa	d: 0.0 Done: False) d: 0.0 Done: False ght)						
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	d: 0.0 Done: False ft) d: 0.0 Done: False						
FFF FHFH FFFH FFFG FFFH FFFH FFFH FFFH	d: 0.0 Done: False ft) d: 0.0 Done: False						
FHFH FFFH AFFF FHFH FFFH AFFG Rewa (U) FFFF	d: 0.0 Done: True d: 0.0 Done: False						
FFFH FFFH FFFG REWAL FFFF FFFH FFFG REWAL FFFF FFFF	d: 0.0 Done: False ght) d: 0.0 Done: False ft)						
FFH HFFG CHAP FFFH HFFG CHAP FFFH FFFH FFFF FFFH FFFF	d: 0.0 Done: False) d: 0.0 Done: False ft)						
FFFH HFFG REWA FFFF FHFH FFFH HFFG REWA	d: 0.0 Done: True d: 0.0 Done: False ght)						
HFFG Rewa (U) FFFH FFFH FFFH KFFG Rewa FFFF FFFH FFFH FFFH FFFH	d: 0.0 Done: False ght)						
HFFG Rewa GFFF FFFH HFFG Rewa GFFF FFFH HFFG FFFH HFFG	d: 0.0 Done: False ght) d: 0.0 Done: False ft)						
HFFG Rewa SFFF FHFH FFFH HFFG Rewa SFFF FHFH FFFH HFFG FFFH	d: 0.0 Done: False ft) d: 0.0 Done: False ft) d: 0.0 Done: True						
Rewaller FFF FFF FFF FFF FFF FFF FFF FFF FFF F	d: 0.0 Done: True d: 0.0 Done: False wn) d: 0.0 Done: True						
FFF FHFH FFFH HFFG (ewa. (R FFF FFFH FFFH	d: 0.0 Done: True d: 0.0 Done: False ght) d: 0.0 Done: False						
(Logical Control Contr	ft) d: 0.0 Done: False						
(D) FFF FHFH FFFH KFFG FFFF FFFH FFFH KFFG Rewa	d: 0.0 Done: False wn) d: 0.0 Done: False ft) d: 0.0 Done: False ght)						
(R. FFFF FHFH FFFH HFFG EFFF FHFH FFFH HFFG (D.							
(D) SFFF FHFH FFFH AFFG Rewa Lo SFFF FHFH FFFH FFFH FFFH REWa							
(D) SFFF FHFH FFFH HFFG Rewa (U) SFFF	d: 0.0 Done: False wn) d: 0.0 Done: False						
FHFH FFFH AFFG Rewa (U) FFFF FHFH FFFH AFFG Rewa	d: 0.0 Done: False) d: 0.0 Done: False ght)						
HFH FFFH KEWA CFFF HFH FFFH KFFG CEWA CFFF	d: 0.0 Done: False wn) d: 0.0 Done: True						
(Do FFF FHFH FFFH HFFG Rewa (R SFFF HFH FFFH	d: 0.0 Done: False wn) d: 0.0 Done: False ght)						
FFFH HFFG Rewa FFFFH FFFH HFFG Rewa U FFF FHFH FFFH FFFH	d: 0.0 Done: False ft) d: 0.0 Done: False						
FFFH HFFG Rewa FFFFH FFFH HFFG Rewa (R FFFF FFFH FFFFH FFFFH FFFH	d: 0.0 Done: False ft) d: 0.0 Done: False ght)						
FFFH HFFG Rewa (R FFFF FHFH FFFH HFFG Rewa	d: 0.0 Done: False ght) d: 0.0 Done: False ft)						
HFFG Rewa (R SFFF HFH FFFH HFFG Rewa SFFF FHFH FFFH HFFG	d: 0.0 Done: False ght) d: 0.0 Done: False d: 0.0 Done: False						
Rewal (Li FFF FHFH FFFH HFFG Rewal (U) FFF FHFH FFFH HFFG	d: 0.0 Done: False ft) d: 0.0 Done: False) d: 0.0 Done: False						
Rewa (R SFFF FHFH HFFG Rewa (D SFFF FHFH FFFH	d: 0.0 Done: False ght) d: 0.0 Done: False wn) d: 0.0 Done: True						
Q3. Q You w provid	Learning [50 points] I implement Q-learning to ed below: f q_learning (env,	alpha=0.5, ging for the gizeros.	gamma=0.95, given envir environmen	epsilon=0.1, onment.			he function
: : : :	earam alpha: Learning aram gamma: Decay in param num_episodes: Seturn: Q table, i.e	ng rate param rate (future Number of eg e. a table wi g was described i Q-learn Initialize Repeat (for Initialize Repeat	neter. reward dis pisodes to ith the Q vi n lecture, but the Q vi $Q(s,a), \forall s \in S, a \in A(s)$ or each episode): the S (for each step of episo	count) parame use for learn alue for ever for your reference TD control algorithm), arbitrarily, and Q(termin	ing. Ey $\langle S, A \rangle$ pair. E, we provide it here $a_{al-state, -1} = 0$		
	_learning(env, alph p.random.seed(42) init method Z: zero f init_method == "Z Q = np.zeros((en	Repeat Cho Take Q(S) S + until S na=0.5, gamma= os R: random Z": nv.nS, env.nA	(for each step of episoose A from S using pose action A , observe R , A	licy derived from Q (e.g., ϵ S' $+ \gamma \max_a Q(S', a) - Q(S, A)$)]	<pre>it_method = "</pre>	Z", test
):		nv.nS, env.nA "R": and(env.nS, e episodes): .reset()	nv.nA)				
:		:		ace.sample()			
):	<pre>#decrease explor ep = epsilon if ep < .01: ep = .01 else: ep = ep0 while not done: if not test: #epsilor exploreF if exploref curr else: curr else:</pre>	rAction = np.	argmax(Q[cu				
):	<pre>currState = env. #decrease explor ep = epsilon if ep < .01: ep = .01 else: ep = ep0 while not done: if not test: #epsilor exploreF if explor curr else:</pre>	ion = np.argm eward, done, ew = np.max(Q[n	<pre>info= env.s ewState]) Action] +</pre>	step(currActi	on) ward + (gamma	* max_Q_NS_NA	.) - Q[cu

[0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.]	
[0.48534231 0.46878849 0.5194411 0.46408375] [0.46693305 0.46623345 0.53428926 0.46492327] [0.4597266 0.45459603 0.45886645 0.4621456] [0.54172166 0.63341978 0.5374195 0.54177959] [0.61185289 0.13949386 0.29214465 0.36636184] [0.45606998 0.57287079 0.34435014 0.53100679] [0.59241457 0.04645041 0.60754485 0.17052412] [0.6410392 0.63561013 0.79089643 0.63749734] [0.69357723 0.58341088 0.5892822 0.59666224] [0.58239996 0.67727719 0.03438852 0.54598549] [0.25877998 0.66252228 0.31171108 0.52006802] [0.54671028 0.18485446 0.96958463 0.77513282] [0.64598392 0.99467557 0.65601253 0.63832127] [0.0884925 0.53497754 0.04522729 1.31741169] [0.38867729 0.27134903 0.82873751 0.35675333] [0.08867729 0.27134903 0.82873751 0.35675333] [0.08867729 0.27134903 0.82873751 0.35675333] [0.00000000000000000000000000000000000	
nit method: Z [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.]]]]]]]]]]]]]]]]]]]]
[0. 0. 0. 0.] [0. 0. 0. 0.]	
[0. 0. 0. 0.] [0. 0. 0. 0.]] [amber of episodes: 5000 nit method: R [0.55149886 0.49972364 0.58956048 0.51139369] [0.5019539 0.55037547 0.51137665 0.4884029] [0.49755446 0.49940424 0.56201374 0.48718521] [0.49948128 0.49786965 0.47708786 0.48720137] [0.60952613 0.59946972 0.55395699 0.54418431] [0.61185289 0.13949386 0.29214465 0.36636184] [0.54576453 0.59215651 0.57336714 0.55281758] [0.59241457 0.04645041 0.60754485 0.17052412] [0.74844574 0.89733871 0.75182424 0.76541 0.66823128 0.79257804 0.7366075 0.70694899] [0.72597838 0.66738328 0.68268606 0.66080876] [0.25877998 0.66252228 0.31171108 0.52006802] [0.54671028 0.18485446 0.96958463 0.77513282] [0.92422003 1.0530922 0.95290452 0.90753825]]]]]]]]]]]]]]]]]]]]
[0.92422003 1.0530922 0.95290452 0.90753825] [1.04055451 1.06960608 1.022507 1.02052069] [0.38867729 0.27134903 0.82873751 0.35675333] [amber of episodes: 5000] [o.it method: Z [2.44776422e-01 1.58550390e-01 1.46705311e-03] [3.62909384e-02 3.05865793e-02 8.42972829e-03] [7.18724342e-02 1.06183150e-01 6.39533410e-03] [8.08323891e-02 3.24316328e-02 3.37429735e-03] [8.08323891e-02 3.24316328e-02 3.37429735e-03] [0.00000000e+00 0.00000000e+00 0.00000000e+00] [2.60933576e-04 9.23698514e-03 3.55434595e-03] [0.00000000e+00 0.00000000e+00 0.00000000e+00] [3.21882348e-02 1.88176798e-01 2.10721882e-03] [1.85117678e-01 5.71450154e-01 4.45026648e-03] [6.80686521e-01 3.11099023e-01 9.99508053e-03] [0.000000000e+00 0.00000000e+00 0.00000000e+00]]]] 1 1.45451936e-01] 2 1.54388567e-01] 2 9.37587624e-02] 2 1.04562198e-01] 1 9.99638558e-02] 0 0.00000000e+00] 2 1.74299107e-04] 0 0.00000000e+00] 1 5.65585828e-01] 1 8.67083240e-02] 2 1.40278735e-01] 0 0.00000000e+00]
[0.00000000e+00 0.00000000e+00 0.00000000e+00 [4.52113168e-01 9.92110944e-02 6.79193944e-02 [4.35912359e-01 9.19237974e-01 5.90198884e-03 [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.0000000000	<pre>0 0.00000000e+00] 1 4.49637599e-01] 1 5.71196276e-01] 0 0.0000000e+00]]]]]]]]]]]]]]]]]]]</pre>
[0.61460671 0.73549539 0.6346824 0.70378371] [0.25877998 0.66252228 0.31171108 0.52006802] [0.54671028 0.18485446 0.96958463 0.77513282] [0.92268756 0.9004001 0.8053061 0.89926124] [0.95960485 1.00098089 1.00206902 1.5957106] [0.38867729 0.27134903 0.82873751 0.35675333] [omber of episodes: 10000 [omit method: Z [o. 0. 0. 0.]]]]]]]
[0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0.]]]]]
[0.59241457 0.04645041 0.60754485 0.17052412] [0.74635231 0.87628864 0.69598811 0.76433875] [0.62327062 0.93606169 0.78117807 0.67838585] [0.60595169 0.73565028 0.65804726 0.65856106] [0.25877998 0.66252228 0.31171108 0.52006802] [0.54671028 0.18485446 0.96958463 0.77513282] [0.90304095 0.92992973 1.24833157 0.90462978] [1.08943457 1.34291473 1.15088582 1.15656326] [0.38867729 0.27134903 0.82873751 0.35675333] [0.mber of episodes: 50000 [0.14131438 0.12029907 0.12893783 0.13076668] [0.080137 0.07158801 0.031121 0.1193003] [0.05906959 0.06063423 0.06290004 0.07302855] [0.04793487 0.0245233 0.01004081 0.07275349] [0.22076694 0.06297318 0.06436951 0.03814453] [0. 0. 0. 0.	
[0.01132289 0.00589803 0.10605686 0.00887796] [0.]]]]]]]]]]]
 Only submit original code written entirely by you. Permitted non-standard libraries: gym, numpy. Only use numpy for random sampling, and seed at the Unwrap the OpenAl gym before providing them to the env = gym.make("FrozenLake-v0").unwrapped Appendix nis appendix includes references to APIs you may find useful ocumentation for a much better description than we could pumpy 	ese functions. ful. If the description sounds useful, check out the respective packag
 np.zeros: N-dimensional tensor initialized to all 0s. np.ones: N-dimensional tensor initialized to all 1s. np.eye: N-dimensional tensor initialized to a diagonal material np.random.choice: Randomly sample from a list, allowing np.argmax: Index of the maximum element. np.abs: Absolute value. np.mean: Average across dimensions. np.sum: Sum across dimensions. PenAl Gym Environment (unwrapped): env.nS # Number of spaces. env.nA # Number of actions. 	
env.P # Dynamics model.	

In [5]: numE = [500, 1000, 5000, 10000, 50000]
 initMethod = ["R", "Z"]

for method in initMethod:

for numEps in numE:

env = gym.make('FrozenLake-v0').unwrapped

print("init method:", method)

print("number of episodes:", numEps)