Name: Ye TIAN   studentID: z5032449

Q1
(a)
(2)
top cycle {a, b, c, d, e, f}

find a cycle {a, b, c}, a can not beat e
expand the cycle to {a, b, c, d, e, f} , every points in this cycle beats every points outside the cycle

(1)
Uncovered set {a, b, c}
Uncovered set is a subset(perhaps proper) of top cycle

(3)
Copeland winner   CO(T) = {c}

Copeland score of a = 4
Copeland score of b = 4
Copeland score of c = 5
Copeland score of d = 3
Copeland score of e = 3
Copeland score of f = 2
Copeland score of g = 0

(4)
banks winners {c}

(5)
Condorcet winners: {c}

a dominate b , d, g and f
b dominate c, d, e and g
c dominate a, d, g, f and e
d dominate e, f and g
e dominate a, f and g
f dominate b and g
g dominate nothing

(b)
Pure Nash equilibria:
assume player 1 can choose A or B, player 2 can choose D or E
If player 1 choose A, player 2 would choose E for higher reward
If player 1 choose B, player 2 would choose D for higher reward
If player 2 choose D, player 1 would choose B for higher reward
If player 2 choose E, player 1 would choose A for higher reward
so, there are two pure Nash equilibria (A,E) and (B,D) refer to (8,5) and (6,6) in the table

Mixed Nash equilibria:
assume player 1 choose A with probability x, choose B with probability 1-x
player 2 choose D with probability y, choose E with probability 1-y

for player 1
$2x + 6(1-x) = 8x + 4(1-x) \implies x = 1/4$

for player 2
$u(D) = u(E)$
$4y + 5(1-y) = 6y + 4(1-y) \implies y = 1/3$

given the pure Nash equilibria (8,5) and (6,6)
the mixed strategies are (2, 5/3) and (9/2, 4)

Q2
(a)
Blackjack: (D) POMDP
Candy Crush: (E) None/Other
Chess: (E) None
Minesweeper: (D) POMDP
Snakes and Ladders: (A) Markov process
Texas Hold'em Poker: (E) None

(b)
When discount factor is very high, the discount effect become meaningless, the total rewards is similar to a linear function. π*(s1) = S, π*(s2) = S (because if choose leave, it is a high risk to go to s3 which value would be always negative), π*(s3) = S or L (because same rewards).

(c)
When discount factor is very low, the immediate rewards will be dominant for the total rewards. π*(s1) = S, π*(s2) = L, π*(s3) = S or L (because same rewards).

(d)
v0(s1) = v0(s2) = v0(s3) = 0

v0(s1, S) = u(s1, S) + δ * P(s1, S, s1) * v0(s1) = 1 + 0.6*1*0 = 1
v0(s2, S) = u(s2, S) + δ * P(s2, S, s2) * v0(s2) = 0 + 0.6*1*0 = 0
v0(s3, S) = u(s3, S) + δ * P(s3, S, s3) * v0(s3) = -2 + 0.6*1*0 = -2

v0(s1, L) = u(s1, L) + δ * P(s2, L, s1) * v0(s2) = 0 + 0.6*1*0 = 0
v0(s2, L) = u(s2, L) + δ * [P(s2, L, s1) * v0(s1) + P(s2, L, s3) * v0(s3)]=5 + 0.6*(0.5*0 + 0.5*0) = 5
v0(s3, L) = u(s3, L) + δ * P(s3, L, s3) * v0(s3) = -2 + 0.6*1*0 = -2

v1(s1) = max[v0(s1,S), v0(s1, L)] = 1
v1(s2) = max[v0(s2,S), v0(s2, L)] = 5
v1(s3) = max[v0(s3,S), v0(s3, L)] = -2

v1(s1, S) = u(s1, S) + δ * P(s1, S, s1) * v1(s1) = 1 + 0.6*1*1 = 1.6
v1(s2, S) = u(s2, S) + δ * P(s2, S, s2) * v1(s2) = 0 + 0.6*1*5 = 3
v1(s3, S) = u(s3, S) + δ * P(s3, S, s3) * v1(s3) = -2 + 0.6*1*-2 = -3.2

v1(s1, L) = u(s1, L) + δ * P(s1, L, s2) * v1(s2) = 0 + 0.6*1*5 = 3
v1(s2, L) = u(s2, L) + δ * [P(s2, L, s1) * v1(s1) + P(s2, L, s3) * v1(s3)]=5 + 0.6*(0.5*1 + 0.5*-2) = 4.7
v1(s3, L) = u(s3, L) + δ * P(s3, L, s3) * v1(s3) = -2 + 0.6*1*-2 = -3.2

v2(s1) = max[v1(s1,S), v1(s1, L)] = 3
v2(s2) = max[v1(s2,S), v1(s2, L)] = 4.7
v2(s3) = max[v1(s3,S), v1(s3, L)] = -3.2

v2(s1, S) = u(s1, S) + δ * P(s1, S, s1) * v2(s1) = 1 + 0.6*1*3 = 2.8
v2(s2, S) = u(s2, S) + δ * P(s2, S, s2) * v2(s2) = 0 + 0.6*1*4.7 = 2.82
v2(s3, S) = u(s3, S) + δ * P(s3, S, s3) * v2(s3) = -2 + 0.6*1*-3.2 = -3.92

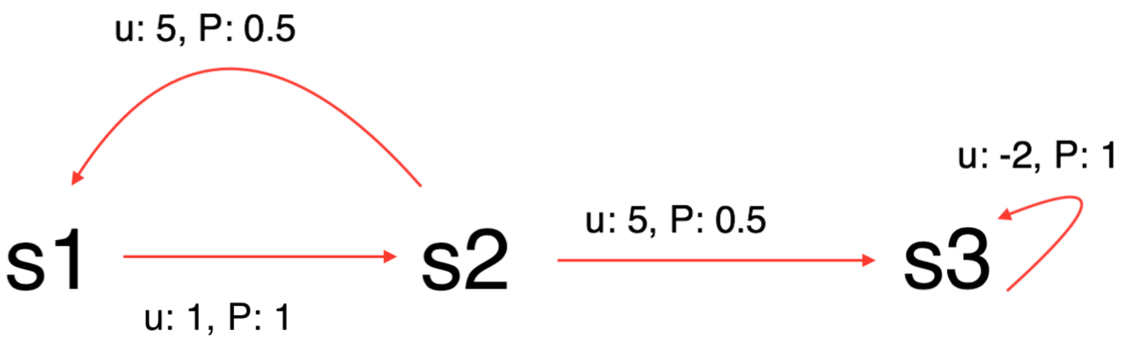
v2(s1, L) = u(s1, L) + δ * P(s1, L, s2) * v2(s2) = 0 + 0.6*1*4.7 = 0.96
v2(s2, L) = u(s2, L) + δ * [P(s2, L, s1) * v2(s1) + P(s2, L, s3) * v2(s3)]=5 + 0.6*(0.5*3 + 0.5*-3.2) = 4.94
v2(s3, L) = u(s3, L) + δ * P(s3, L, s3) * v2(s3) = -2 + 0.6*1*-3.2 = -3.92

v3(s1) = max[v2(s1,S), v2(s1, L)] = 2.8

v3(s2) = max[v2(s2,S), v2(s2, L)] = 4.94
v3(s3) = max[v2(s3,S), v2(s3, L)] = -3.92

	v0(s)	v0(s, S)	v0(s, L)	v1(s)	v1(s, S)	v1(s, L)	v2(s)	v2(s, S)	v2(s, L)	v3(s)
s1	0	1	0	1	1.6	3	3	2.8	0.96	2.8
s2	0	0	5	5	3	4.7	4.7	2.82	2.94	4.94
s3	0	-2	-2	-2	-3.2	-3.2	-3.2	-3.92	-3.92	-3.92

(e) Markov Chain:



u: Utility   P: probability

(f)

V(s1) = 0 + δ x V(s2)
V(s2) = 5 + δ x [0.5 x v(s1) + 0.5 x v(s3)]
V(s3) = -2 - 2 x δ - 2 x δ^2 - …

(1- δ) V(s3) = -2 ==> V(s3) = $\dfrac{2}{\delta - 1}$

V(s2) = 5 + δ x [0.5 x δ x V(s2) + 0.5 x $\dfrac{2}{\delta - 1}$] ==> v(s2) = $\dfrac{(5 - 6\delta)(1 + \delta)}{2}$

v(s1) = $\dfrac{\delta(5 - 6\delta)(1 + \delta)}{2}$

For s3, because for stay and leave, the reward and state after execute action are same. So, no matter the δ value is, the policy is stay or leave.
For s2, if δ is large, under given policy V(s2) $\approx$ -1. However, in this case, if choose stay, V(s2) = 0. So, if δ is large, π^*(s2) = S. If δ is small, under given policy V(s2) $\approx$ 5. If choose stay, V(s2) = 0. So, if δ is small, π^*(s2) = L.

For s1, if δ is large, under given policy V(s1) = δ x V(s2) $\approx$ -1. However, if choose stay, under given policy V(s1) = $1 + \delta^2 + \delta^3 + \ldots$ = $\dfrac{1}{1-\delta}$ = 1000. So, if δ is large, π*(s1) = S. If δ is small, under given policy V(s1) = δ x V(s2) $\approx$ 0. However, if choose stay V(s1) $\approx$ 1. So, if δ is small, π*(s2) = S.