Ravina Lad.

EX3 → Dynamic Programming.

Q.1

a) $V_*(s) = \sum_a \pi^*(a|s)\, q_*(s,a)$

b) $q^*(s,a) = \sum_{s',r} p(s',r|s,a)\,(r + \gamma V_*(s'))$

c) $\pi_*(s) = \arg\max_a q_*(s,a)$

d) $\pi_*(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)(r + \gamma V_*(s'))$

e) Bellman eq^n (V_π, V_*, q_π, q_*) in terms of 3 args^{tr} p & 2 args f^n r.

$$V_*(s) = \max_a \left[r(s,a) + \gamma \sum_{s'} p(s'|s,a)\, V_*(s') \right]$$

$$V_\pi(s) = \sum_a \pi(a|s) \left[r(s,a) + \sum_{s'} p(s'|s,a) \cdot \gamma V_\pi(s') \right]$$

$$q_\pi(s,a) = r(s,a) + \sum_{s'} p(s'|s,a) \left[\gamma \sum_a \pi(a'|s')\, q_\pi(s' \right.$$

$$q_*(s,a) = r(s,a) + \sum_{s'} p(s'|s,a)\, \gamma \max_{a'} q_*(s',a')$$

Q.2 Fixing Policy iteration –

a) We can fix the bug in policy iteration by every time selecting optimal actions with equal probability.

eg – if policy has 4 action $a = \{$ Left, Right, Up, down $\}$ at each iteration will select optimal policy will equal probability. if Left & Up are optimal then →

$$\pi(a) = \{0.5, 0, 0.5, 0\}$$

Here's modified pseudo code →

1. Initialization – $V(s) \in R \quad \forall s \in S$
   $\pi(s) \in A(s)$
2. Policy evaluation.

loop → $\Delta \leftarrow 0$
loop for each $s \in S$
$v \leftarrow V(s)$
$V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} P(s', r | s, a)[r + \gamma V(s')]$
$\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small +ve number)

3. Policy improvement
policy-stable ← True
for each $s \in S$:
old-action ← $\pi(s)$
$\pi(s) \leftarrow \frac{1}{\text{No. of optimal actions.}} \cdot \text{argmax}_a \sum_{s',r} P(s', r | s, a)[r + \gamma V(s')]$
if old-action $\neq \pi(s)$:
policy stable ← False.
if policy-stable:
stop & return $V \approx V_*$ & $\pi \approx \pi_*$
(Break)
else:
go to 2.

b) • I don't think so that value iteration has any bug similar to policy iteration.
• I think optimal value fn should be unique.
• It should also unique regardless of how many different optimal policies we have.

Q.3) Policy Iterations for action values:

a) 1. Initialization:

$q(s,a) \in \mathbb{R}$.
$\pi(s) \in A(s)$ arbitarily $\forall s \in S$.

2. Policy Evaluation:

Loop:
$\Delta \leftarrow 0$
Loop for each $s \in S$:
loop for each $a \in A$:

$$q(s,a) \leftarrow \sum_{s',r} p(s',r|s,a)\,[r + \gamma \sum_{a'} \pi(a'|s')\, q(s',a')]$$

$$\Delta \leftarrow \max(\Delta; \text{abs}(q(s,a) - q(s,a)))$$

until $\Delta <$ θ (small positive number)

3. Policy Improvement:

policy-stable $\leftarrow$ True
For each $s \in S$: for each $a \in A$:
old-action $\leftarrow \pi(s)$
$\pi(s) \leftarrow \underset{a}{\text{argmax}}\, q(s,a)$
If old.action $\neq \pi(s)$:
policy-stable $\leftarrow$ False
If policy-stable, stop & return $q \approx q^*$ & $\pi \approx \pi^*$
else go to 2.

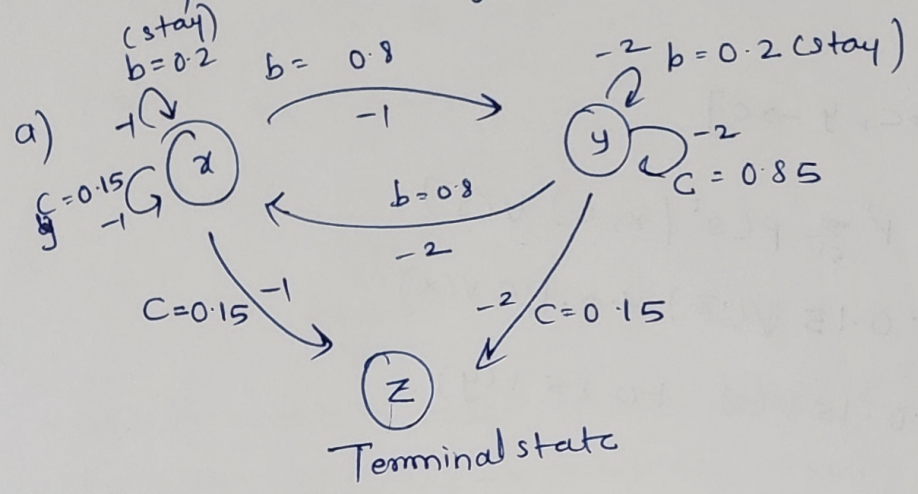
If $\left(\sum_{s',r} p(s',r|s,\pi(s))[r + \gamma \sum_{a'} \pi(a'|s')\, q(a',s')\right) \neq \sum_{s',r} p(s',r|s,\text{old-action})$
$[r + \gamma \sum_{a'} \pi(a'|s') \times q(a'|s)]$

b)

$$q_{k+1}(s,a) \leftarrow \sum_{s',r} p(s',r|s,a)\left[r + \gamma \max_{a'} q_k(s',a')\right]$$

Q4 Policy - iteration by hand,

a)



- agent's goal is to reach → z → Terminal state
- from any state x / y → prob → Terminal state = c = 0.15
- considering reward values,

  State x → receive -1

  State y → receive -2

- To minimize -ve rewards → I think agent should stay at x.
- optimal policy would be -

  ① starting in y → move to x → take c from x → Goal

  OR

  ② starting in x → move to c → reach goal

(B) iteration 1

$V(x) = -1 + 0.15\, V(z) + 0.85\, V(x)$ (with $V(z) = 0$)

$V(y) = -2 + 0.2\, V(y) + 0.8\, V(x)$.

$V(z) = 0$

$V(x) = \frac{-1}{0.15} = \frac{-20}{3} = -6.67$

$0.8\, V(y) = -2 - 0.8\left(\frac{20}{3}\right)$

$V(y) = \frac{-7.33}{0.8} = -9.1625$

Policy improvement

$\pi(x) = \underset{a}{\text{argmax}} \begin{cases} -1 + 0.8\, V(y) + 0.2\, V(x) \\ -1 + (0.15 \times 0) + 0.85\, (V(x)) \end{cases}$

$= \underset{a}{\text{argmax}} \begin{cases} -1 + \overset{-7.3}{0.8(-9.16)} + \overset{-1.33}{0.2(-6.67)} \\ -1 + 0.85(-6.67) \end{cases}$

$= \underset{a}{\text{argmax}} \begin{cases} -9.63 \\ -6.67 \end{cases}$

$\pi(x) = c$

$\pi(y) = \underset{a}{\text{argmax}} \begin{cases} -2 + 0.8\, (V(x) + 0.2\, V(y)) \\ -2 + 0.85\, V(y) \end{cases}$

$= \underset{a}{\text{argmax}} \begin{cases} -2 + \overset{-5.53}{0.8(-6.67)} + \overset{-1.82}{0.2(-9.16)} \\ -2 + 0.85(-9.16) \end{cases}$

$= \begin{cases} -8.156 \\ -9.786 \end{cases}$

$\pi = [x \to c,\ y \to b] \to$ same as prev. policy is converged.

c) Initial policy $\pi = [x \rightarrow b, y \rightarrow b]$

$V(x) = -1 + 0.8\, V(y) + 0.2\, V(x)$

$V(y) = -2 + 0.8\, V(x) + 0.2\, V(y)$

$V(z) = 0$

$0.8\, V(x) = -1 + 0.8\, V(y)$

$0.8\, V(y) = -2 + 0.8\, V(x)$

$0.8\, V(x) = -1 + [-2 + 0.8\, V(x)]$

$0 = -3$ which is inconsitent.

Our initial policy always takes action b which causes it to loop bet^n stat x & y ∞.

Heres we are taking undiscounted MDP,

will get sum of ∞ -ve no.

• Does discounting help?

→ discounting factor can help in this case.
As values in future will not be significant.

→ it will be difficult to acheive optimal solution as problem changes. Also, MDP optimal policy is independent of discounting factor, policy might be able to converge.

Q.7) Proving convergence of value iteration.
for any f^n f & g.

$$|\max_a f(a) - \max_a g(a)| \leq \max_a |f(a) - g(a)|$$

Hint: consider the cases $\max_a f(a) - \max_a g(a) \geq 0$.
& $\max_a f(a) - \max_a g(a) < 0$.

→ ① case 1 →

$$\max_a f(a) - \max_a g(a) \geq 0. \quad -①$$

Let, $\max_a f(a) = a^*_f$, $\max_a g(a) = a^*_g$

$$f(a^*_f) - g(a^*_g) \geq 0.$$

$$\therefore |f(a^*_f) - g(a^*_g)| \leq \max_a |f(a) - g(a)| \quad -②$$

$g(a^*_f) \leq g(a^*_g)$ → a^*_g → optimal action for g.

from eqn ①,

$$f(a^*_f) - g(a^*_f) \geq f(a^*_f) - g(a^*_g) \geq 0.$$

$$f(a^*_f) - g(a^*_g) = |f(a^*_f) - g(a^*_f)|$$

$$f(a^*_f) - g(a^*_g) < f(a^*_f) - g(a^*_f)$$

$$= |f(a^*_f) - g(a^*_g)| \leq \max_a |f(a) - g(a)|$$

2nd case →

$\max_a f(a) - \max_a g(a) = f(a^*_f) - g(a^*_g) < 0$

$f(a^*_g) \leq f(a^*_f)$

$\rightarrow g(a^*_g) - f(a^*_g) \geq g(a^*_g) - f(a^*_f) > 0$

$g(a^*_g) - f(a^*_g) = |g(a^*_g) - f(a^*_g)|$

$\therefore g(a^*_g) - f(a^*_f) \leq g(a^*_g) - f(a^*_g)$

$= |g(a^*_g) - f(a^*_g)| \leq \max_a |g(a) - f(a)|$

$= \max_a |f(a) - g(a)|$

b) $\left| \max_a f(a) - \max_a g(a) \right| \leq \max_a |f(a) - g(a)|$

$V_k \leftarrow$ 12th value iteration

$B \leftarrow$ Bellman Backup operator

$V_{k+1} \leftarrow BV_k$

$V_{k+1}(s) \leftarrow \max_a \sum p(s', r | s, a)[r + \gamma V_k(s')]$

$||V||_\infty$ max absolute value of it's element.

→ from given defination ~~title~~ in the problem,

$$\|\beta V_i - \beta V_i'\|_\infty = \| R(s) + \max_a \gamma \sum p(s'|s,a) V(s') - R(s) - \max_a \gamma \sum p(s'|s,a) V'(s') \|_\infty$$

$$= \| \max_a \gamma \sum p(s'|s,a) \cdot V(s') - \max_a \gamma \sum p(s'|s,a) V'(s') \|$$

→

$$\leq \max_a | \gamma \sum p(s'|s,a) V(s') - \gamma \sum p(s'|s,a) V'(s') |$$

$$\leq \max_a \gamma \sum p(s'|s,a) \| V - V' \|$$

$$= \max_a \gamma \| V - V' \|$$

$$\sum_{s'} p(s'|s,a) = 1$$

$$\therefore \max_a \gamma \| V - V' \| = \gamma \| V - V' \|$$

$$\therefore \| \beta V_i - \beta V_i' \|_\infty \leq \gamma \| V_i - V_i' \|_\infty$$