سوال ۱)

الف)

حالت خاصی از فرآیندهای تصادفی که خروجی وابسته به State و Action فعلی است و گذشته در نتیجه عملی که در زمان حال صورت می‌گیرد تاثیری ندارد (خاصیت Memory less).

Actionهای ما در MDP دارای عدم قطعیت هستند؛ در نتیجه ما به جای Planning دنبال Policy هستیم.

ب)

بله سودوکو یک فرآیند مارکوفی است زیرا حالت‌های گذشته و آینده از هم مستقل هستند و خروجی تنها به حالت و اکشن واسته است. خیر مارکوفی نیست زیرا باید از وضعیت حالت‌ها و اکشن‌ها قبلی باخیر باشیم (بدانیم چه سوالاتی پرسیده بودیم.)

پ)

تمرکز اصلی policy iteration بر روی ارزیابی policy ها می‌باشد در حالی که value iteration حالت‌ها و جفت حالت-اکشن را ارزیابی می‌کند و policy را به دست می‌آورد.

محدودیت‌های value iteration:

1. هر دور هزینه‌ای معادل $O(|S|^2|A|)$ دارد که کمی پر هزینه می‌باشد.
2. احتمالش هست که هزینه مربوط به برخی حالت‌ها تغییر نکند اما به دلیل تغییر در تعداد کمی از حالت‌ها الگوریتم باید به کار خودش ادامه دهد.
3. در بعضی موارد policy به دست آمده به همگرایی رسیده است اما مقادیر (Values) در value iteration هنوز به همگرایی نرسیده‌اند که باعث می‌شود فرآیند ادامه داشته باشد و عامل اتلاف وقت شود.

یک نقطه ضعف درمورد policy iteration این است که در هر دور ارزیابی policy نیز انجام می‌شود که بار محاسباتی زیادی دارد.

در مجموع policy iteration زودتر از value iteration به همگرایی می‌رسد.

ت)

خیر MDP هایی که سلف لوپ دارن و نمیشه براشون درخت expectimax کشید چون عمق درخت نامحدود می‌شود.

سوال ۲)

الف)

در هر دو حالت ما اطلاعاتی درباره transitions و rewards نداریم. اما در passive هدف ما به دست آوردن ارزش هر state می‌باشد ولی در active می‌خواهیم optimal valueها و optimal policy ها را یاد بگیریم.

همچنین عامل یادگیرنده در حالت passive انتخابی انجام نمی‌دهد و صرفا policy وارد شده را اجرا می‌کند و نتیجه را مشاهده می‌کند اما در حالت active خود عامل در لحظه انتخاب و تجربه می‌کند.

ب)

این روش یک policy است که بهترین اکشن که بالاترین value را دارد را با احتمال 1-epsilon که بین صفر و یک می‌باشد و یک اکشن رندوم که احتمال epsilon را دارد انتخاب می‌کند. مشکل این روش این است که وقتی اکشن‌های رندوم را انتخاب می‌کند، آن‌ها به صورت uniform انتخاب می‌کند یعنی همه آن‌ها به یک مقدار خوب درنظر می‌گیرد با اینکه برخی از آن‌ها انتخاب‌هایی بهتری می‌توانند باشند.

یک راه‌حل برای این مشکل استفاده از softmax action selection rules می‌باشد که احتمال هر اکشن بر اساس یک graded function محاسبه می‌کند که ارزش حدس‌زده شده را برمی‌گرداند؛ مانند Gibbs و Boltzmann.

پ)

در exploration ما استیت های ناشناخته را تجربه می‌کنیم تا نسبت به محیط تجربه کسب کنیم. وقتی اطلاعات کسب شده به انداز کافی شد حال باید از این دانش کسب شده استفاده کنیم یعنی exploit کنیم تا بتوانیم نهایت استفاده را از reward ها ببریم پس در استخراج به نوعی از تجربه گذشته استفاده میکنیم.

به عنوان مثال می‌توان رانندگی خودکار که در آن عامل باید بین مسیرهایی که قبلا طی کرده و مسیرهای جدید انتخاب کند، شطرنج که در آن عامل باید بین موقعیت‌ها یا شروع بازی‌هایی که قبلا تجربه کرده و موقعیت‌های جدید و ناشناخته انتخاب کند، معاملات سهام که عامل باید بین خرید سهام‌هایی که تجربه خرید آن‌ها را داشته و شرکت‌های جدید و نوپا انتخاب کند، اشاره کرد.

ت)

خیر نمایش با ویژگی تقریبی از تابع q است که لزومی ندارد بتواند مقدار بهینه را برای مقادیر q پیدا کند.

سوال ۳)

الف)

S =	(1, 1)	(1, 2)	(1, 3)	(2, 1)	(2, 2)	(2, 3)
$\pi^*(s)$ =	Up	Left	NA	Right	Right	NA

ب)

S =	**(1, 1)**	**(1, 2)**	**(1, 3)**	**(2, 1)**	**(2, 2)**	**(2, 3)**
$V_0(S)$ =	**0**	**0**	**0**	**0**	**0**	**0**
$V_1(S)$ =	**0**	**0**	**0**	**0**	**4.0**	**0**
$V_2(S)$ =	**0**	**2.38**	**0**	**2.88**	**4.0**	**0**

ج)

عامل باید بتواند که در جهان با انجام دادن اکشن‌ها و دیدن اثرات آن‌ها، کاوش کند.

د)

این بخش حذف شده است.

سوال ۴)

الف)

$$V_0^{\pi}(S) = 0$$

$$V_{K+1}^{\pi}(S) = \sum_{S'} T(S,\pi(S),S')[R(S,\pi(S),S') + \gamma\, V_K^{\pi}(S')]$$

$$V_1^{\pi}(S_3) = 3, V_1^{\pi}(S_4) = 4, V_1^{\pi}(S_5) = 5, V_1^{\pi}(S_6) = 6$$

$$V_1^{\pi}(S_1) = V_1^{\pi}(S_2) = \frac{1}{6}(-1+0) + \frac{1}{6}(-1+0) + \frac{1}{6}(-1+0) + \frac{1}{6}(-1+0) + \frac{1}{6}(-1+0) + \frac{1}{6}(-1+0) = -1$$

$$V_2^{\pi}(S_1) = V_2^{\pi}(S_2) = \frac{1}{6}(-1-1) + \frac{1}{6}(-1-1) + \frac{1}{6}(-1+3) + \frac{1}{6}(-1+4) + \frac{1}{6}(-1+5) + \frac{1}{6}(-1+6) = 1.67$$

$$V_3^{\pi}(S_1) = V_3^{\pi}(S_2) = \frac{1}{6}(-1+1.67) + \frac{1}{6}(-1+1.67) + \frac{1}{6}(-1+3) + \frac{1}{6}(-1+4) + \frac{1}{6}(-1+5) + \frac{1}{6}(-1+6) = 2.56$$

$$V_4^{\pi}(S_1) = V_4^{\pi}(S_2) = \frac{1}{6}(-1+2.56) + \frac{1}{6}(-1+2.56) + \frac{1}{6}(-1+3) + \frac{1}{6}(-1+4) + \frac{1}{6}(-1+5) + \frac{1}{6}(-1+6) = 2.58$$

$$V_5^{\pi}(S_1) = V_5^{\pi}(S_2) = \frac{1}{6}(-1+2.58) + \frac{1}{6}(-1+2.58) + \frac{1}{6}(-1+3) + \frac{1}{6}(-1+4) + \frac{1}{6}(-1+5) + \frac{1}{6}(-1+6) = 2.95 \approx\ 3$$

حالت	S1	S2	S3	S4	S5	S6
π(i)	ریختن تاس	ریختن تاس	اتمام بازی	اتمام بازی	اتمام بازی	اتمام بازی
V(π(i))	**3**	**3**	**3**	**4**	**5**	**6**

ب)

طبق ارزش‌های به دست آمده در بخش الف، پاداش مورد انتظار ریختن تاس برابر است با:

$$\frac{1}{6}(-1+3) + \frac{1}{6}(-1+3) + \frac{1}{6}(-1+3) + \frac{1}{6}(-1+4) + \frac{1}{6}(-1+5) + \frac{1}{6}(-1+6) = 3$$

بنابراین تنها در حالت‌های S1 و S2 بهتر است که تاس ریخته شود:

حالت	S1	S2	S3	S4	S5	S6
π(i)	ریختن تاس	ریختن تاس	اتمام بازی	اتمام بازی	اتمام بازی	اتمام بازی
π(i+1)	**dice**	**dice**	**dice/finish**	**finish**	**finish**	**finish**

ج)

همانطور که در جدول مشخص است، policy در حالت i نسبت به حالت i + 1 تغییر خاصی نکرده است. حتی در حالت ۳ هم هر دو تصویر بهینه هستند پس نمی‌توان از آن به عنوان تغییر policy عنوان کرد. بنابراین می‌توان گفت که policy همگرا یا converge شده است. البته توجه داشته باشید که در حالی که policy بهینه است، اما ممکن است valueها بعد policy همگرا شوند. عموما در مسائل کاربردی، policy زودتر از value ها همگرا می‌شود.

سوال ۵)

الف)

$$\boldsymbol{Q\big((3,2),N\big) = \frac{50+50}{2} = 50}$$
$$\boldsymbol{Q\big((3,2),S\big) = \frac{30+30}{2} = 30}$$
$$\boldsymbol{Q\big((2,2),E\big) = \frac{50+50+30+30}{4} = 40}$$

ب)

حالت اول: از ابتدا ارزش خانه‌های پایانی برابر با پاداششان در نظر گرفته شود:

EP1

$$\boldsymbol{4\!: Q\big((3,2),N\big) = (1-0.5)*0+0.5*(0+50) = 25}$$

EP2

$$\boldsymbol{3\!: Q\big((2,2),S\big) = (1-0.5)*0+0.5*(0-100) = -50}$$

EP3

$$\boldsymbol{3\!: Q\big((3,2),E\big) = (1-0.5)*0+0.5*(0+25) = 12.5}$$
$$\boldsymbol{4\!: Q\big((3,2),S\big) = (1-0.5)*0+0.5*(0+30) = 15}$$

EP4

$$\boldsymbol{2\!: Q\big((1,2),E\big) = (1-0.5)*0+0.5*(0+12.5) = 6.25}$$

.
.
.

$$\boldsymbol{Q\big((1,2),E\big) \to (4,2), Q\big((3,2),E\big) \to (3,3), Q\big((3,2),S\big) \to (3,4)}$$

حالت دوم: از ابتدا ارزش خانه‌های پایانی برابر با صفر در نظر گرفته شود:

EP1

$$5: \boldsymbol{Q\big((3,3),Exit\big) = (1-0.5)*0+0.5*(0+50) = 25}$$

EP2

$$4: \boldsymbol{Q\big((2,1),Exit\big) = (1-0.5)*0+0.5*(0-100) = -50}$$

EP3

$$5: \boldsymbol{Q\big((3,1),Exit\big) = (1-0.5)*0+0.5*(30+0) = 15}$$

EP4

$$4: \boldsymbol{Q\big((3,2),N\big) = (1-0.5)*0+0.5*(0+25) = 12.5}$$

$$5: \boldsymbol{Q\big((3,3),Exit\big) = (1-0.5)*25+0.5*(0+50) = 37.5}$$

EP5

$$3: \boldsymbol{Q\big((2,2),E\big) = (1-0.5)*0+0.5*(0+12.5) = 6.25}$$

$$4: \boldsymbol{Q\big((3,2),S\big) = (1-0.5)*0+0.5*(0+15) = 7.5}$$

$$5: \boldsymbol{Q\big((3,1),Exit\big) = (1-0.5)*15+0.5*(30+0) = 22.5}$$

$$\boldsymbol{Q\big((1,2),E\big) \rightarrow never, Q\big((2,2),E\big) \rightarrow (5,3), Q\big((3,2),S\big) \rightarrow (5,4)}$$

ج)

هر کدام از موارد زیر مورد تایید می‌باشد:

$$Q(s_t,a_t) = (1-\alpha)Q(s_t,a_t) + \alpha(r_t + \gamma r_{t+1} + \gamma^2 max_{a'}Q(s_{t+2},a'))$$

$$Q(s_t,a_t) = (1-\alpha)Q(s_t,a_t) + \alpha(r_t + \gamma((1-\alpha)Q(s_{t+1},a_{t+1}) + \alpha(r_{t+1} + \gamma max_{a'}Q(s_{t+2},a'))))$$

$$Q(s_t,a_t) = (1-\alpha)Q(s_t,a_t) + \alpha(r_t + \gamma \max\,((1-\alpha)Q(s_{t+1},a_{t+1}) + \alpha\big(r_{t+1} + \gamma max_{a'}Q(s_{t+2},a')\big), max_{a'}Q(s_{t+1},a')))$$

سوال ۶)

الف)

فاصله از نزدیک‌ترین روح، فاصله از نزدیک‌ترین غذا، تعداد غذاهای باقی‌مانده و …

ب)

$F_g = 2, F_p = 1$

ج)

$$Q(s,a) = w_g * F_g + w_p * F_p = 100 * 2 + (-10) * 1 = 190$$

د)

$$Q(s',a') = 100 * 0 + (-10) * 2 = -20$$
$$Q_{target}(s,a) = r(s,a) + \gamma Q(s',a') = 50 + 0.5 * (-20) = 40$$

ه)

$$Difference = Q_{target}(s,a) - Q(s,a) = 40 - 190 = -150$$
$$w_g = w_g + \alpha[Difference]F_g = 100 + 0.5 * (-150) * 2 = -50$$
$$w_p = -10 + 0.5 * (-150) * 1 = -85$$

ی)

دور بودن از روح‌ها (تعداد کم) و نزدیک بودن به غذاها (تعداد زیاد) به عنوان ارزش آموخته می‌شود.

اگر W_g مقداری کوچک و W_p مقداری بزرگ به خود بگیرد، احتمالا پکمن عملکرد درستی خواهد داشت.

سوال ۷

الف)

که عامل همیشه به سمت راست برود.

ب)

$$\mathbf{10 + 10 * \frac{1}{2} + 10 * \left(\frac{1}{2}\right)^2 + \cdots = 10 * \frac{1}{1 - {}^{1}\!/_{2}} = 20}$$

ج)

$$\boldsymbol{V^*(n-1) = 1 + \frac{1}{2}V^*(n)}$$

$$\boldsymbol{V^*(n-2) = 1 + \frac{1}{2}V^*(n-1) = 1 * \left(1 + \frac{1}{2}\right) + \left(\frac{1}{2}\right)^2 V^*(n)}$$

$$\boldsymbol{= \frac{1 - (1/2)^2}{1 - 1/2} + \left(\frac{1}{2}\right)^2 V^*(n)}$$

.

.

.

$$\boldsymbol{V^*(n-k) = 1 + \frac{1}{2}V^*(n-k+1) = 1 + \frac{1}{2} + \cdots (\frac{1}{2})^{k-1} + \left(\frac{1}{2}\right)^k V^*(n)}$$

$$\boldsymbol{= \frac{1 - (1/2)^k}{1 - 1/2} + \left(\frac{1}{2}\right)^k V^*(n)}$$

د)

بعد از یک دور می‌دانیم که $\mathbf{V_1(n) = 10}$ و $\mathbf{V_1(n\text{-}k) = 1}$ برای تمام **k**های بزرگ‌تر از صفر. پس:

$$\boldsymbol{V_2(n) = 10 + 10 * 0.5 = 15}$$

$$\boldsymbol{V_2(n-1) = 1 + \frac{1}{2}V_1(n) = 6}$$

$$\boldsymbol{V_2(n-k) = 1 + \frac{1}{2}V_1(n-k+1) = 1 + 1/2 = 3/2 \;\; \forall k > 1}$$