۱-

- سوال ۱: ۱- استفاده از این روش از نظر محاسباتی گران است. زیرا باید محاسبه‌ی کردن مقادیر V تا زمان همگرایی مقادیر (یا در صورت نیاز به محاسبه‌ی کمتر، همگرایی سیاست‌ها) ادامه یابد، که معمولاً زمان زیادی طول می‌کشد.
۲- استفاده از این روش نیازمند ذخیره‌سازی کل مقادیر state-value function است، که می‌تواند در مورد مسائل MDP با اندازه‌ی بزرگ مشکل‌ساز باشد.

۳- تعیین کردن مقدار مناسب برای discount factor یک مسئله‌ی اساسی در استفاده از این روش و رسیدن به سیاست‌های درست و بهینه است، و معمولاً در ابتدا مشخص نیست که باید برای آن چه مقداری را در نظر گرفت. در نظر گرفتن مقداری بد برای آن می‌تواند سبب تولید سیاست‌هایی suboptimal شود.

۲-

- سؤال ۱:بهتر است به ضریب تخفیف دست نزنیم، زیرا مقدار آن مناسب است، به این دلیل که قرار است سیاست ما به گونه‌ای شود که عبور از پل انتخاب شود، و برای این کار باید مقدار امتیازی که با هر بار گام برداشتن و به هدف نرسیدن از reward کلی ما کم می‌شود کم باشد تا عامل رفتن به سمت راست را به رفتن به سمت چپ و زودتر به هدف رسیدن ترجیح دهد. ولی لازم است که مقدار نویز را بسیار کم کنیم که عامل مقدار خطا و احتمال این که با حرکت در مسیر رسیدن به سمت راست پل با انتخاب حرکتی به صورت تصادفی به پایین بیفتد را بسیار کم در نظر بگیرد و سبب شود که عامل همچنان عبور از پل را ترجیح دهد.

- سؤال ۲: در بسیاری از موارد هنگام حل مسائل MDP، در نظر گرفتن یک ضریبِ تخفیف راه خوبی برای رسیدن به سیاست‌های درست است، زیرا در مسائل دنیای واقعی معمولاً رسیدن به پاداش‌های کوتاه‌مدت و در زمان کمتر به رسیدن به پاداش‌هایی در زمان دورتر اولویت دارد. پس با استفاده از این ضریب، و در واقع با اعمال تخفیفی روی جمع rewardی که عامل در کل می‌گیرد با هر حرکت که منجر به رسیدن به حالت پایانی نشده است، می‌توان این عامل را نیز به مدل‌سازی اضافه کرد و در نتیجه به نتایج بهتری رسید.

- سؤال ۳: بله. می‌توان برای حل این مسأله از Policy Iteration به جای آن استفاده کرد. در این روش، ما از دو متود Policy evaluation و Policy Extraction استفاده می‌کنیم. در روش اول، به گرفتن یک سیاست در ابتدای مسئله، مقادیر value را برای حالت‌ها محاسبه می‌کنیم، و در روش دوم، با گرفتن مقادیر حالت‌ها، برای آن‌ها سیاست‌های بهینه را استخراج می‌کنیم. حال با ترکیب این دو روش به این صورت که ابتدا با تعیین یک سیاست دلخواه روش اول را تا همگرا شدن مقادیر ادامه دهیم، و سپس برای این مقادیر با استفاده از روش دوم سیاست جایگزین که بهتر از سیاست قبلی است را طراحی می‌کنیم و این کار را ادامه می‌دهیم. از لحاظ پیچیدگی زمانی، Policy Iteration دارای پیچیدگی زمانی از مرتبه‌ی AN^2 است که در آن N تعداد حالات و A تعداد اکشن‌ها در هر حالت است، و در روش دوم پیچیدگی زمانی N^3 است، ولی معمولاً روش دوم به تعداد پیمایش‌های کمتری برای همگرا شدن نیاز دارد، و همین سبب کم‌تر شدن زمان آن در کل نسبت به روش اول می‌شود.

-۳

- سؤال ۱:

a) چون خروجی نزدیک را ترجیح داده‌ایم، پس باید ضریب تخفیف مقداری نسبتاً کم باشد. همچنین به دلیل پذیرفتن ریسک صخره، پس پاداشی که با هر حرکت گرفته‌ایم مقدار منفی نسبتاً بزرگی بوده است که سبب شده است مسیر کوتاه‌تر انتخاب کنیم. همچنین مقدار نویز نیز باید کم بوده باشد تا ریسک گذشتن از مسیر خطرناک‌تر را بپذیریم.

b) به دلیل قبلی، ضریب تخفیف همچنان باید مقدار کمی باشد. اما با وجود جریمه‌ی زیاد، به دلیل این که نویز در این مسأله مقدار زیادی بوده است، عامل ریسک گذشتن از مسیر کوتاه‌تر ولی پرخطرتر را نمی‌پذیرد و مسیر امن‌تر را انتخاب می‌کند.

c) به دلیل دور بودن خروجی انتخاب شده، پس ضریب تخفیف باید زیاد و نزدیک به یک باشد. همچنین به دلیل پذیرفتن ریسک صخره، پس باید نویز کمی داشته باشیم و همچنین مقدار پاداش هر حرکت نیز باید مقداری منفی باشد.

d) به دلیل قسمت قبل، باید ضریب تخفیف زیاد و نزدیک به یک باشد. ولی به دلیل اجتناب از صخره، باید نویز زیاد باشد تا عامل ریسک رد شدن از مسیر کوتاه‌تر را با وجود پاداش منفی‌ای که می‌گیرد نپذیرد و مسیر بلندتر ولی امن‌تر را انتخاب کند.

e) در صورتی که ضریب تخفیف را نزدیک به یک بگیریم و همچنین مقدار نویز را نیز زیاد، و از همه مهم‌تر مقدار پاداشی که برای هر حرکت می‌گیریم را نیز مقداری مثبت و نسبتاً بزرگ در نظر بگیریم، عامل بهترین سیاست را فقط در حرکت کردن و نرسیده به هیچ حالت نهایی و در نتیجه افزایش امتیازش در نظر می‌گیرد. به همین دلیل به سمت بالا رفتن و باز هم به این کار ادامه می‌دهد.

- سؤال ۲: می‌توان با کاهش پاداش زنده ماندن عامل، و همچنین کم کردن ضریب تخفیف عامل را وادار به پایان بازی کرد.

- سؤال ۳: لزوماً خیر. تنها زمانی که گاما عددی بین صفر و یک باشد موجب به همگرایی می‌شود، زیرا ضرب پی‌درپی این مقدار سبب کم شدن مقادیر اضافه شده در هر گام و در نتیجه همگرایی می‌شود.

-۴

سؤال ۱:

روش batch :

حسن: مقادیر state values زودتر همگرا می‌شوند و همچنین دقیق‌تر هستند.
عیب:نیاز به زمان و محاسبه‌ی زیادی است.

روش تکی:
حسن:با صرف زمان و محاسبه‌های کمتری عامل به دید نسبتاً قابل قبولی می‌رسد.
نکته‌ی منفی: دیرتر همگرا شده و همچنین دقت کم‌تری دارد.

۶-

سؤال ۱) در صورتی که مقدار Q برای اقداماتی که عامل قبلاً ندیده است زیاد باشد، عامل تمایل بیش‌تری به explore کردن محیط دارد، و درواقع عامل به سراغ امتحان کردن اکشن‌هایی که تا به حال امتحان نکرده است می‌رود، و اگر این مقدار کم باشد، عامل تمایل بیش‌تری به exploit کردن محیط دارد و درواقع عامل به سراغ انجام کارهایی که تا الان انجام داده و امن‌تر هستند می‌رود.

سؤال ۲) این الگوریتم یک الگوریتم off-policy است، زیرا این الگوریتم مقادیر تابع action-value راصرف نظر از اکشنی که در حال حاضر انجام شده است آپدیت می‌کند. این به این معنی است که عامل با استفاده از این الگوریتم سیاست بهینه را یاد می‌گیرد، حتی اگر به صورت نیمه‌بهینه عمل کند و محیط را explore کند. همچنین این الگوریتم value-based است زیرا عامل درباره‌ی مقادیر بهینه‌ی q-valueها تجربه کرده و می‌آموزد، و نه به صورت مستقیم در مورد خود سیاست بهینه برای هر حالت.

سؤال ۳) TD-learning از به روز رسانی تفاوت زمانی (TD) برای تخمین تابع ارزش بر اساس تفاوت بین تخمین فعلی ارزش یک حالت و تخمین ارزش حالت بعدی به اضافه پاداش فوری استفاده می کند. TD-learning می تواند تخمین های خود را پس از هر مرحله زمانی به روز کند و نیازی به تکمیل یک episode به صورت کامل ندارد. این باعث می شود که برای یادگیری آنلاین در محیط هایی که عامل فقط به اطلاعات جزئی دسترسی دارد مناسب باشد.

در مقابل، روش‌های MC تابع ارزش را با میانگین‌گیری پاداش‌های واقعی مشاهده‌شده در چندین قسمت کامل تخمین می‌زنند. روش‌های MC نیازمند تکمیل episode کامل قبل از انجام هر گونه به‌روزرسانی برای تابع مقدار هستند. این باعث می شود که آنها برای یادگیری آفلاین در محیط هایی که عامل به اطلاعات کامل در مورد محیط دسترسی دارد، مناسب باشند.

یکی از مزیت های یادگیری TD نسبت به روش های MC این است که می تواند از اطلاعات ناقص یا تا حدی قابل مشاهده یاد بگیرد. TD-learning به عامل اجازه می دهد تا پاداش مورد انتظار آینده را بر اساس پاداش مشاهده شده و برآورد فعلی ارزش حالت بعدی تخمین بزند. در مقابل، روش‌های MC به قسمت‌های کامل تجربه قبل از انجام هر گونه به‌روزرسانی نیاز دارند.

از سوی دیگر، روش‌های MC می‌توانند نسبت به یادگیری TD از نظر نمونه کارآمدتر باشند، به ویژه در شرایطی که فضای حالت بزرگ و پیچیده است. این به این دلیل است که روش‌های MC به طور مستقیم پاداش مورد انتظار را برای هر حالت با میانگین‌گیری در بسیاری از مسیرهای ممکن تخمین می‌زنند، در حالی که یادگیری TD به بوت استرپ از برآوردهای فعلی خود از تابع ارزش متکی است.

۷-

هدف از این کار این است که همگام با این که سیاست بهینه را دنبال کرده و به دنبال بهبود سیاست و عمل کردن آن است، هر بار با یک احتمالی نیز دست به امتحان کردن حرکات شانسی بزند و درواقع با یک احتمالی نیز محیط را explore کند تا در صورتی که به اکشن‌هایی رسید که منجر به امتیاز نهایی بهتری می‌شوند، آن‌ها را نیز لحاظ کند. این کار سبب می‌شود که تعادل خوبی بین دنبال کردن سیاست بهینه‌ی یافت‌شده و همچنین امتحان کردن راه‌های تصادفی که ممکن است منجر به بهبود شوند برقرار شود.

۸-
سؤال ۱) برای دستیابی به سیاست بهینه، تعداد ۵۰ اپیزود تعداد بسیار کمی است و به تعداد بیش‌تری نیاز داریم، با توجه به این‌که در این‌جا سیاست بهینه نیز در واقع عبور از پل است.

سؤال ۲) افزایش اپسیلون موجب به تمایل بیش‌تر عامل به explore کردن محیط می‌شود و درواقع ترجیح می‌دهد که اکشن‌های جدید بیش‌تری را امتحان کند و نتیجه‌ی آن‌ها را ببیند. همچنین کاهش اپسیلون نیز منجر به تمایل عامل به exploit کردن محیط می‌شود و عامل ترجیح می‌دهد کارهایی که از قبل تجربه کرده است و تا آن زمان بهینه بوده است را انجام بدهد.

۹-
در این بخش، ما در واقع Q-value ها را به صورت ترکیبی خطی از مجموعه‌ای از فیچرها تعریف می‌کنیم، زیرا نگه داشتن همه‌ی Q-value ها به صورت جداگانه در مسائل دارای ابعاد بزرگ از لحاظ حافظه‌ای بسیار هزینه‌بر است. پس در واقع برای هر فیچر در معادله‌ی Q-value یک وزن به دست می‌آوریم و در آخر این مقدار را با قرار دادن فیچرها در معادله به دست می‌اوریم. مقادیر Q-value هر بار با استفاده از فرمول‌های آورده شده از اسلایدهای درس آپدیت می‌شوند که به طور خلاصه با استفاده از آپدیت هر وزن در هر مرحله با استفاده از مقدار قبلی خود و تجربه‌ی جدید به دست‌آمده است.