۱.سوال : حداقل ۳ مورد از مشکلات روش Iteration Value را نام برده و توضیح دهید.

۱.پیچیدگی محاسباتی: الگوریتم باید هی تابع ارزش را بررسی کند برای هر جفت موقعیت و حرکتی تا وقتی که همگرا شود. این کار میتواند زمان زیادی را بگیرد بخصوص در مسئله های با موقعیت های زیاد

۲. مشکلات با بعد مسئله، با افزایش حالات و حرکات، پیچیدگی محاسباتی بطور نمایی افزایش میابد. باعث میشود ارزیابی همه ی جفت های حالت-حرکت بسیار دشوار یا ناممکن شود.

۳.همگرایی : گاهی ممکن است این همگرایی رخ ندهد یا خیلی کند باشد.

۲.سوال: دلیل انتخاب این مقادیر را به زبان ساده و به صورت شهودی توضیح دهید.

۱. اگر نویز را 0 کنیم هیچگاه تصادفا به -۱۰ نمیرود و خطری ما را تهدید نمیکند.

۲. با تخفیف بیشتر، پاداش های کم و نزدیک اولویت بیشتری میگیرند تا پاداش های دور و زیاد.

۳. چون policy iteration هزینه ی محاسباتی کمتری نسبت به value iteration دارد.

۳.سوال: چگونه به این نتیجه رسیدیم که در حل مسائل با روش Iteration Value از Factor Discount استفاده میکنیم و این فاکتور چه کمکی به ما میکند؟(میتوانید از نتایجی که در این بخش گرفتید نیز کمک بگیرید و به کمک آن ها توضیح دهید).

برای برقراری تعادل بین جایزه های کنونی و اینده استفاده میشود. باعث میشود جایزه های کنونی نسبت به جایزه های آینده اولویت بگیرند. با این روش، iteration value میتواند همگرا شود.

۴.سوال: راه حل Iteration Value راهی زمانبر است که باید برای هر State همه حالتها را بسنجیم و گاهی ناگزیر به انجام آن هستیم. اما در این مسئله به خصوص آیا راه حل سادهتری نسبت به Iteration Value وجود دارد که تعداد حالتهای بررسی شده را کاهش دهد؟ این روش را نام ببرید و توضیح دهید و سپس آن ها را از نظر پیچیدگی زمانی مقایسه کنید.

۵.سوال: دلیل انتخاب خود برای هریک از مقادیر پارامترهای مذکور را در هر سیاست بیان کنید.

برای ترجیح +۱ به +۱۰ باید جایزه زندگی را منفی تر و میزان تخفیف را بیشتر کنیم

برای ترجیح راه پر خطر باید نویز را کمتر کنیم.

۶.سوال: در سیاست پنجم، همانطور که مشاهده کردید در یک لوپ بینهایت میافتادیم و عامل علاقه ای به پایان بازی نداشت. برای حل این مشکل چه راه حل هایی به نظرتان میرسد. آن ها را توضیح دهید.

۱. تعیین حداکثر تعداد مراحل: پس از تعداد مشخصی مرحله الگوریتم متوقف شود حتی اگر همگرا نشده باشد

۲. معیار متوقف کردن: اگر تفاوت بین مقادیر قدیم و جدید از مقدار مشخصی کمتر باشد، الگوریتم متوقف شود.

۳. تصادفی عمل کردن. اگر برخی حرکات را تصادفی انجام دهیم، ممکن است از حلقه ی بینهایت بیرون بیاییم.

۷.سوال: آیا استفاده از الگوریتم تکرار ارزش تحت هر شرایطی به همگرایی میانجامد؟

بله اما با شروطی. فاکتور تخفیف باید کمتر از ۱ باشد. همچنین تعداد حالات و حرکات باید قابل شمارش باشند، بینهایت نباشند.

۸.سوال: روشهای بروزرسانیای که در بخش اول (بروزرسانی با استفاده از batch)و در این بخش (بروزرسانی به صورت تکی) پیاده کرده اید را با یکدیگر مقایسه کنید. (یک نکته مثبت و یک نکته منفی برای هرکدام)

دسته ای: تخمین دقیق تری از مقدار واقعی به ما میدهد. اما تاخیر در آپدیت داریم و همگرایی را کند تر میکند

تکی: نیاز به فضای حافظه ی کمی دارد. چون بر اساس هر تجربه، آپدیت میشود، خطای بیشتری دارد.

۹.سوال: توضیح دهید که اگر مقدار Q برای اقداماتی که عامل قبلا ندیده، بسیار کم یا بسیار زیاد باشد چه اتفاقی میافتد.

باید تصمیمگیری غیربهینه یا حتی نادرست میشود. اگه زیادی بزرگ باشد، عامل به برداشتن آن اقدام ادامه میدهد حتی اگر در واقعیت بهینه نباشد. اگر هم خیلی کم باشد، عامل آن را انتخاب نمیکند حتی اگر راه بهینه باشد.

۱۰.سوال: بیان کنید learning-Q یک الگوریتم policy-Offاست یا policy-On؟ value-based است یا policy-based؟ توضیح دهید.

Q-learning is a value-based policy-off reinforcement learning algorithm.

این به این معنا است که در هنگام یادگیری، یک سیاست را دنبال نمیکند بلکه عمل بهینه را با تابع کیو یاد میگیرد. این الگوریتم از معادله ی بلمن استفاده میکند تا مقادیر کیو را بر اساس جایزه های گرفته شده برای هر حرکت در حالت مشخص، آپدیت کند.

۱۱.سوال: الگوریتم leaning-Q از Leaning-TD استفاده میکند آن را با Carlo Monte مقایسه کنید و بیان کنید استفاده هر کدام چه مزایا و چه معایبی دارند.

در TD-Learning تابع کیو با استفاده از جایزه ی پیشبینی شده و تجربیات نصفه آپدیت میشود. اما در monte carlo تجربیات کامل هستند و پیشبینی ای در کار نیست.

خوبی TD-learning این است که سریع تر حالات را پوشش میدهد اما محاسبات بیشتری میخواهد.

خوبی monte carlo این است که نیاز به حدس و گمان ندارد اما نمیتواند در لحظه و آنلاین عمل کند.

۱۲.سوال: هدف از استفاده از اپسیلون و به کارگیری روش اپسیلون حریصانه چیست؟

هدف ایجاد تعادل بین کاوش و بهره برداری است. در فرآیند تصمیم گیری، در صورتی که همیشه بهترین گزینه را انتخاب نکند، با احتمال کمی، یک گزینه تصادفی را انتخاب میکند و به کاوش روش های جدید و جایگزین های بهتر ادامه میدهد. این الگوریتم، در مواقعی که عامل در محیطی نامعلوم باید قرار بگیرد، استفاده ی خوبی دارد.

۱۳.سوال: حال، همین کار را با اپسیلون ۰ دوباره تکرار کنید. آیا مقدار اپسیلون و ضریب یادگیری ای وجود دارد که با استفاده از آنها، سیاست بهینه با احتمال خیلی بالا (بیشتر ۹۹ درصد) بعد از ۵۰ بار تکرار یاد گرفته شود؟

۱۴.سوال: به صورت ساده و شهودی توضیح دهید که با کم یا زیاد کردن مقدار epsilon روند یادگیری عامل چگونه تغییر می کند.

افزایش اپسیلون باعث افزایش حرکات تصادفی و کشف راه های جدید میشود اما کاهشش موجب استفاده ی بیشتر از دانش کنونی میشود. بهتر است در اوایل یادگیری اپسیلون بزرگ و سپس کم بشود.

۱۵.سوال: تغییرات و فعالیت هایی که در این بخش انجام داده اید را توضیح دهید.