1. Jelaskan cara kerja dari algoritma Q-Learning dan SARSA!
Jawab:

1) Q-Learning

   Q-Learning adalah algoritma Reinforcement Learning yang digunakan untuk mempelajari kebijakan optimal dari sebuah agen yang berinteraksi dengan lingkungan untuk mencapai tujuan tertentu. Algoritma ini berusaha memaksimalkan nilai kumulatif dari reward yang diterima oleh agen. Cara kerjanya adalah sebagai berikut:

   a) Tabel Q diinisialisasi dengan nilai nol atau acak. Tabel ini akan berisi nilai-nilai Q untuk setiap kombinasi state dan action, yang merupakan representasi estimasi reward yang akan diterima agen jika memilih suatu aksi pada suatu state tertentu.

   b) Agen akan menggunakan strategi epsilon-greedy untuk menentukan apakah agen akan melakukan eksplorasi (mencoba aksi baru) atau eksploitasi (memanfaatkan pengalaman yang sudah didapatkan).

      i) Eksplorasi: Dengan probabilitas epsilon, agen memilih aksi secara acak untuk mengeksplorasi state dan action yang belum pernah dipilih.

      ii) Eksploitasi: Dengan probabilitas 1-epsilon, agen akan memilih aksi dengan nilai Q terbesar berdasarkan pengalaman sebelumnya.

   c) Saat agen berada di sebuah state dan memilih sebuah action, agen akan pindah ke state berikutnya dan menerima reward. Nilai Q diperbarui menggunakan formula Q-Learning update rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

   - $Q(s,a)$ adalah nilai Q sebelum diperbarui.
   - α (learning rate) menentukan seberapa besar pembelajaran baru mempengaruhi nilai Q lama.
   - r adalah reward yang diterima dari lingkungan setelah melakukan aksi.
   - γ (discount factor) adalah faktor diskon untuk reward di masa depan. Nilai γ berkisar antara 0 dan 1. Semakin dekat dengan 1, semakin penting nilai dari reward di masa depan.
   - $\max_{a'} Q(s',a')$ adalah nilai Q maksimal yang diprediksi untuk state berikutnya s', yang dihitung dari semua aksi yang mungkin diambil dari state tersebut.

   d) Agen akan berinteraksi dengan lingkungan dalam episode yang berulang-ulang. Setiap episode biasanya dimulai dari kondisi awal dan berakhir ketika agen mencapai state terminal (misalnya, menang atau kalah dalam game). Setiap kali agen bergerak, ia memperbarui nilai Q berdasarkan reward yang diterima dan state yang dicapai. Seiring berjalannya waktu dan interaksi yang lebih banyak, agen akan mulai mengeksploitasi lebih banyak karena nilai-nilai Q di tabel akan lebih akurat dalam mencerminkan hasil terbaik untuk setiap state-action pair.

e) Setelah proses pembelajaran selesai, agen akan memiliki Q-table yang berisi nilai terbaik untuk setiap pasangan state-action. Agen dapat memilih aksi yang memberikan reward maksimal di setiap state, yang membentuk kebijakan optimal. Kebijakan optimal adalah strategi di mana agen akan selalu memilih aksi yang memberikan nilai Q tertinggi di setiap state.

2) SARSA

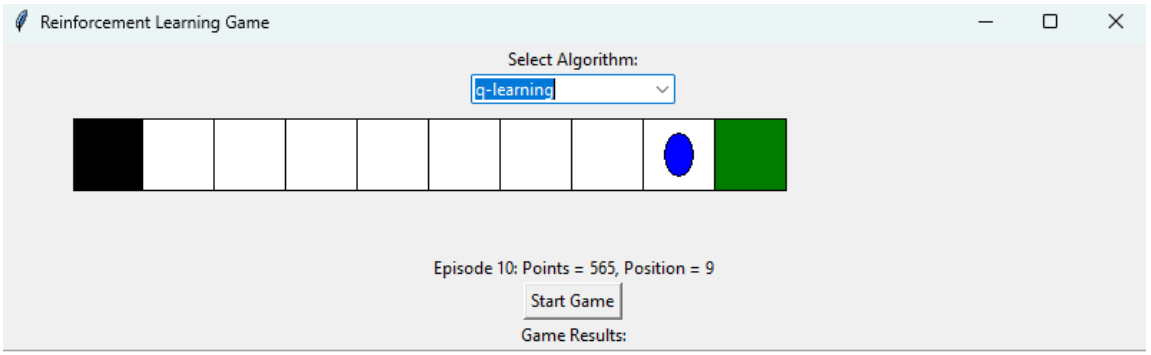
SARSA (State-Action-Reward-State-Action) adalah algoritma Reinforcement Learning berbasis on-policy, di mana agen mempelajari kebijakan optimal sambil mengikuti kebijakan yang sedang dieksekusi. Tujuan SARSA sama dengan Q-Learning, yaitu menemukan kebijakan optimal untuk memaksimalkan reward jangka panjang, tetapi cara memperbarui nilai-nilai Q-nya berbeda. Langkah-langkahnya juga hampir sama dengan algoritma Q-Learning. Yang berbeda hanyalah:

- SARSA memperbarui nilai Q berdasarkan aksi sebenarnya yang diambil di state berikutnya, sedangkan Q-Learning memperbarui nilai Q berdasarkan aksi optimal yang diambil dari state berikutnya.
- SARSA lebih bersifat on-policy, karena agen mengikuti kebijakan yang sedang dipelajari selama pembaruan.

2. Bandingkan hasil dari kedua algoritma tersebut, bagaimana hasil perbandingannya? Jika ada perbedaan, jelaskan alasannya!

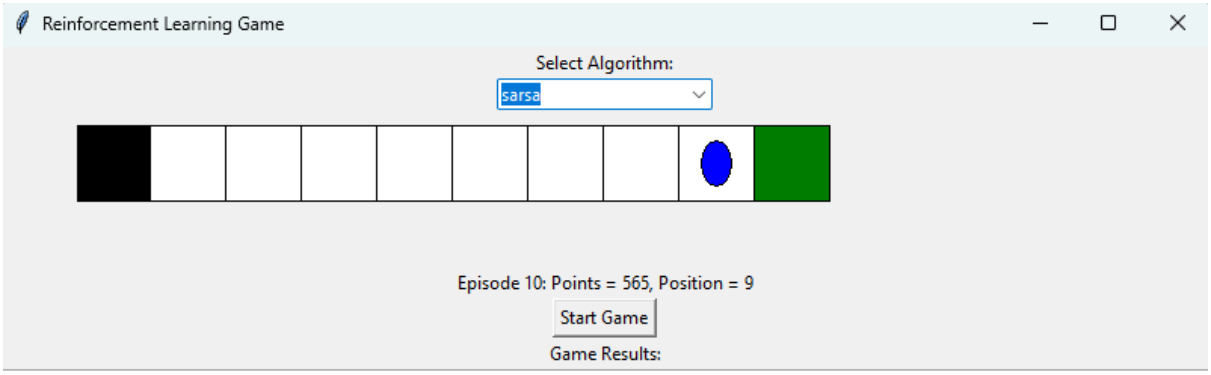
Jawab:

Q-Learning




```
Player won in episode 1 with 538 points. Steps taken: 173
Player won in episode 2 with 537 points. Steps taken: 69
Player won in episode 3 with 555 points. Steps taken: 51
Player won in episode 4 with 553 points. Steps taken: 53
Player won in episode 5 with 569 points. Steps taken: 37
Player won in episode 6 with 565 points. Steps taken: 41
Player won in episode 7 with 567 points. Steps taken: 39
Player won in episode 8 with 565 points. Steps taken: 41
Player won in episode 9 with 565 points. Steps taken: 41
Player won in episode 10 with 565 points. Steps taken: 41
Final Q-Table:
[[-1.00000000e-01 -1.72947595e-01]
 [-1.90000000e+01 -2.09221474e+00]
 [-2.08429591e+00  4.09742216e-01]
 [-1.72213887e+00  1.34140912e+01]
 [-1.08121843e+00  5.18762541e+01]
 [ 1.18869813e+01  6.64970481e+01]
 [ 9.88455987e+00  8.24126546e+01]
 [ 1.17700914e+01  1.00620203e+02]
 [ 1.50742031e+01  1.21794494e+02]
 [-1.37285990e+00  3.36845963e+01]]
```

SARSA




```
Player lost in episode 1 with -211 points. Steps taken: 13
Player won in episode 2 with 552 points. Steps taken: 155
Player won in episode 3 with 547 points. Steps taken: 59
Player won in episode 4 with 550 points. Steps taken: 58
Player won in episode 5 with 557 points. Steps taken: 49
Player won in episode 6 with 565 points. Steps taken: 41
Player won in episode 7 with 565 points. Steps taken: 41
Player won in episode 8 with 563 points. Steps taken: 43
Player won in episode 9 with 563 points. Steps taken: 43
Player won in episode 10 with 565 points. Steps taken: 41
Final Q-Table:
[[-1.00000000e-01 -1.40245382e-01]
 [-2.71000000e+01 -2.00796830e+00]
 [-3.60802841e+00 -9.27192935e-04]
 [-1.85881456e+00  1.17331669e+01]
 [-6.42211424e-01  4.13612341e+01]
 [ 3.39362115e+00  5.63038728e+01]
 [ 6.07707917e+00  6.95906375e+01]
 [ 7.68295621e+00  9.26030022e+01]
 [ 1.84358608e+01  1.11157430e+02]
 [-1.44465474e+00  2.04661954e+01]]
```

- Q-Learning terlihat konvergen lebih cepat dalam jumlah langkah setelah beberapa episode. Misalnya, pada episode pertama butuh 173 langkah, namun episode berikutnya mulai stabil pada sekitar 41 langkah. Hasil konvergensi menunjukkan Q-Learning lebih cepat mencapai hasil optimal dalam hal langkah minimum yang diperlukan untuk mencapai tujuan.
- SARSA membutuhkan lebih banyak langkah di awal episode, misalnya 155 langkah di episode kedua, namun juga menunjukkan stabilisasi pada langkah 41 mulai dari episode ke-7. SARSA tampak lebih lambat untuk mencapai stabilisasi dibandingkan dengan Q-Learning.

Hal ini dikarenakan Q-Learning memperbarui nilai berdasarkan ekspektasi terbaik dari state berikutnya tanpa mempertimbangkan apakah aksi tersebut diambil atau tidak. Oleh karena itu, ia dapat belajar lebih cepat jika nilai Q sudah mendekati optimal. Namun, jika ada noise dalam lingkungan, performanya bisa lebih fluktuatif. Sedangkan, SARSA memperhitungkan aksi aktual yang diambil agen, sehingga pembaruannya lebih lambat tetapi cenderung lebih aman dalam hal stabilitas dan menghindari eksploitasi terlalu dini.