

Respuestas a las preguntas

- a) ¿Qué estructura de datos subyacente se utiliza comúnmente para implementar una cola de prioridad en Python?

Se implementa comúnmente mediante un montículo binario (min-heap), utilizando el módulo `heapq` de Python.

- b) ¿Cuál es la diferencia entre una cola FIFO tradicional y una cola de prioridad?

En una cola FIFO (First-In, First-Out), los elementos se procesan en el orden en que se insertan. En cambio, en una cola de prioridad, los elementos se procesan según su prioridad, sin importar el orden de llegada.

¿Qué módulo proporciona Python para trabajar fácilmente con colas de prioridad?

El módulo `heapq`

- c) ¿Qué ventajas tiene el uso de una cola de prioridad sobre una lista ordenada manualmente?

Una cola de prioridad basada en heap permite inserciones y extracciones eficientes $O(\log n)$, mientras que mantener una lista ordenada manualmente implica costos mayores $O(n)$ para insertar en la posición correcta.

- d) Si dos elementos tienen la misma prioridad, ¿cómo decide la cola cuál atender primero?

Si dos elementos tienen la misma prioridad, se atienden en el orden en que fueron insertados (orden de llegada) si no hay conflicto en comparación. Alternativamente, se puede usar un segundo criterio como un contador para desempatar.

- e) ¿Qué se debe hacer para que los elementos personalizados puedan ser almacenados en una cola de prioridad en Python?

Para almacenar elementos personalizados sin usar programación orientada a objetos, se deben representar como tuplas donde el primer elemento sea la prioridad (un número), y los demás elementos sirvan como criterio de desempate. Por ejemplo, se puede usar una tupla del tipo (prioridad, identificador, datos), donde identificador es un número que asegura un orden claro cuando dos elementos tienen la misma prioridad.

- g) ¿Qué situaciones del mundo real se pueden modelar con colas de prioridad? Mencione al menos dos.

Asignación de órganos para trasplante según urgencia.

Atención en salas de urgencias médicas según la gravedad del paciente.

- h) En un sistema de atención médica, ¿cómo se puede usar una cola de prioridad para organizar a los pacientes?

Se asigna una prioridad más alta a los pacientes más graves, de modo que la cola atienda primero a quienes presentan mayor urgencia médica, independientemente de su orden de llegada.

i) ¿Cómo afectaría al rendimiento usar una lista simple en lugar de una estructura especializada como heapq para manejar prioridades?

El rendimiento sería menor, ya que insertar o buscar el elemento de mayor prioridad en una lista simple toma tiempo $O(n)$ en lugar de $O(\log n)$ que ofrece un heap.

j) ¿Qué complejidad tiene la inserción y extracción en una cola de prioridad basada en heap?

Es preciso decir que tanto la inserción como la extracción del elemento de mayor prioridad en un heap tienen complejidad $O(\log n)$.