

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

Se hace uso de un heap binario, específicamente un mini-heap, como el que implementa el módulo `heapq` de Python.

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

Una cola FIFO (First in First out), elimina o obtiene los elementos teniendo en cuenta únicamente el orden de llegada de estos, los primeros que entran son los primeros que salen. Sin embargo, en una cola de prioridad atiende a los elementos con mayor prioridad, sin importar el orden en el que hayan sido agregados.

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

El módulo que proporciona Python para trabajar fácilmente con colas de prioridad es `heapq`.

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

Una de las ventajas de la cola de prioridad es su eficiencia, ya que las operaciones de inserción y extracción en un heap tienen complejidad $O(\log n)$. Por otro lado, no es necesario mantener manualmente el orden de la lista tras cada operación. De igual forma se puede acceder al elemento de mayor prioridad en $O(1)$.

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

Si dos elementos tienen la misma prioridad, la cola decide atender al que primero fue insertado como si se tratara de una cola tradicional.

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

Para que los elementos personalizados queden almacenados en una cola de prioridad estos deben ser convertidos a tuplas cuyos datos sean la prioridad y el objeto.

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

Se pueden implementar en el sector de la salud, en el área de atención al paciente, priorizándolos según la gravedad de su condición. De igual forma dentro del sistema operativo del cualquier computador se ejecutan ciertos procesos según su prioridad.

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

Se puede asignar a cada paciente una prioridad numérica basada en la gravedad de su condición. Pacientes con condiciones graves tienen mayor prioridad por ende el menor número si se hace uso de heapq y son atendidos primero.

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

Insertar y mantener ordenada una lista tiene un costo de $O(n)$, mientras que heapq permite inserciones y extracciones en $O(\log n)$.

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

La complejidad que tiene la cola de prioridad basada en heap tanto para extracción del máximo o mínimo así como para la inserción es de $O(\log n)$. Por otro lado, el acceso al mínimo o al máximo si extraer es de $O(1)$.