

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

Un binary Heap

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

En una FIFO los elementos se procesan en el mismo orden en que llegan al igual que en una queue. En cambio, una cola de prioridad ordena los elementos justamente según su prioridad, atendiendo primero a aquellos elementos de mayor prioridad sin tener en cuenta el orden de llegada.

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

El modulo heapq

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

El uso de una cola de prioridad permite insertar y eliminar elementos mucho más eficientemente, pues tiene una complejidad de $O(\log n)$. Mientras que, por el otro lado, una lista ordenada manualmente suele requerir más tiempo al tener una complejidad de $O(n)$.

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

La cola los atenderá en el orden en que fueron insertados, funcionando así como una cola FIFO para elementos que tengan la misma prioridad.

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

Deben de tener una "variable" que sirva de comparación, para que a través de una función de criterio de comparación sea capaz de definir la prioridad de cada elemento.

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

En un parque de diversiones donde aquellos que compran un Fast Pass pasan primero a la atracción. En envíos de pedidos por internet, donde aquellos que deciden pagar un envío prioritario o un envío gratis, tienen prioridad en el despacho de sus productos.

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

Se puede organizar a los pacientes por la gravedad de su problema o la urgencia de ser atendido. El sistema decidiría a qué paciente atender primero en base a que tanta atención y urgencia requiera el paciente. Por ejemplo, el sistema prioriza un pre-infarto antes que una gripe, o un derrame cerebral antes que una fractura.

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

Usar una lista simple para manejar prioridades en vez de un heapq implica insertar cada elemento manualmente en una posición de acuerdo con su prioridad, lo que consume más tiempo al tener una complejidad de $O(n)$, lo que haría más ineficiente y menos automatizado el proceso.

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

Tanto la inserción como la extracción tienen una complejidad de $O(\log n)$ en una estructura de heap.