

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

- Para implementar una cola de prioridad se pueden usar árboles o array lists, pero normalmente se usan los árboles.

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

- Una cola FIFO tradicional organiza a los datos por su orden de entrada, mientras que una cola de prioridad organiza los datos en base a un orden asignado, y se organiza en base a su prioridad.

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

- Python utiliza el módulo heapq, que es de bajo nivel y el módulo queue que es de alto nivel.

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

- La complejidad, mientras que en un array tienes que ir recorriendo todas las posiciones (en el peor caso) hasta que en criterio de complejidad se cumpla y volver a recorrerlo hasta que se ordenen todos los datos, en el heap ya tienes la prioridad dada por un $2n \log n$.

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

- si dos elementos tienen la misma prioridad, se va a atender dependiendo de un orden relativo.

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

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

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

i) ¿Qué complejidad temporal tiene la inserción y eliminación/atención en una cola de prioridad basada en heap?