

Daniel Rincon 202420898
Nicolas Castano 202420324
Juan Ecvheverry 202420422

PREGUNTAS:

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

- Al implementar colas de prioridad las estructuras de datos que utilizamos son los heap

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

- FIFO (First in, first out) atiende los elementos en el orden que llegan. Por otro lado, los heaps o colas de prioridad atienden a los elementos según prioridad, no por orden de llegada.

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

- Heapq

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

- Cuando tenemos una lista ordenada y queremos insertar un elemento aleatorio, tenemos que volver a ordenar la lista cada vez que insertamos un elemento. Por otro lado, en un heap las inserciones y extracciones más rápidas ($O(\log n)$), y menor consumo de recursos en comparación con ordenar manualmente usando `.sort`.

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

- Si tienen la misma prioridad, la cola de prioridad atenderá primero al que fue insertado antes (preserva el orden de llegada si las prioridades son iguales).

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

- Se deben hacer comparables (definir `lt`- lista) o usar tuplas donde el primer elemento sea la prioridad y el segundo el objeto. Ejemplo: (prioridad, objeto).

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

- 1) Sistema de atención médica (donde los pacientes que están en caso crítico son atendidos primero)
- 2) Ejecución de las tareas del sistema operativo de un computador

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

Se asigna a cada paciente una prioridad basada en la gravedad de su condición. El sistema extrae y atiende primero al paciente con mayor prioridad, no al que llegó primero.

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

Una lista simple requiere ordenarla con cada inserción o extracción ($O(n \log n)$), lo cual es menos eficiente que heapq, que opera en $O(\log n)$.

j) ¿Qué complejidad tiene la inserción y extracción en una cola de prioridad basada en heap $O(\log n)$.