

Observaciones Lab9

Luis Alejandro Rodríguez Arenas, la.rodrigueza12@uniandes.edu.co, Código 202321287.
Santiago Rojas, s.rojasg23@uniandes.edu.co, Código 202420928

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

Un heap binario

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

FIFO siempre tiene el mismo orden el primero que entra es el primero que sale, en colas de prioridad este atiende el que tenga mayor prioridad, aunque haya llegado mucho después

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

Heap_pq

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

No tienes que volver a ordenar la lista cada vez que entra una nueva persona con diferente prioridad, ya que el heap mantiene el orden de las prioridades automáticamente

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

Si los dos tienen la misma prioridad ahí si se usa FIFO entre ellos

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

Urgencias, después del triage, el de más urgencia atienden primero

En el computador el CPU elige hacer los procesos mas importantes primero

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

asignas prioridad según gravedad el heap asegura que un paciente crítico salte a la cabeza de la cola, aunque haya llegado después.

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

Tendrías que buscar en toda la lista dónde meter el nuevo elemento u ordenar siempre eso termina siendo $O(n)$

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

Insertar y eliminar son de complejidad $O(\log n)$, eliminar el que tiene más prioridad igualmente tiene la complejidad $O(\log n)$.