

Tomas Aponte 202420148

Juan Diego García 202423575

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

Los heap se para implementar una cola de prioridad. Esta estructura es útil ya que hace un árbol binario el cual esta organizado de manera que los hijos de la raíz o de un nodo siempre serán mayores o menores dependiendo de su implementación. Esto facilita la organización de los datos según el retorno requerido.

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

Mientras que, en una cola FIFO el primer dato que llega es también el primer dato que sale, en una cola de prioridad se establece un orden de salida para todos los elementos.

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

Python provee el módulo llamado `heapq` para colas de prioridad, este implementa una estructura heap ya diseñada para las colas de prioridad y evita tener que hacerla manualmente.

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

La mayor ventaja de usar una cola de prioridad, es su baja complejidad temporal al ser comparada con una organizada manualmente. Es mucho mas eficiente que tener que organizar una lista varias veces, debido a que inserta y remueve elementos rápidamente dentro de una estructura organizada.

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

Cuando dos elementos tienen la misma prioridad, usualmente el programa atendería en primer lugar al que llego primero, ya que, no hay ningún otro criterio para decidir el nivel de urgencia.

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

Se podrían guardar los elementos como tuplas con su respectiva prioridad y dato para permitir datos personalizados en una cola, de modo que se organicen por un criterio establecido y guarden a su vez la información necesaria.

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

Las emergencias de un hospital es uno de los ejemplos mas claros de una cola de prioridad. Esto se debe a que, siempre se busca atender al paciente que mas lo necesita primero y en el caso de haber dos casos similares se atenderá al primero que llegó. Otro ejemplo parecido podría ser un esquema de trabajo de una empresa, en el cual los problemas más grandes o complejos son a los que se dedican primero los empleados y su prioridad bajara conforme lo importante que sean.

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

Podría establecerse un sistema numérico, por ejemplo, del 1 al 10 para calificar la prioridad que tiene un paciente frente a los otros. Después se podría almacenar una tupla la prioridad y nombre del paciente. De esta manera el heapq podrá organizar a los pacientes según su prioridad y guardar su nombre para asegurar su atención correspondiente.

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

En una lista simple, cada vez que se inserta un nuevo elemento, seria necesario buscarle un lugar respectivo en la lista, por lo que, se volvería muy ineficiente ordenarla cada vez que se inserte un dato. En cambio, en un heapq siempre se sabrá la posición del elemento con mayor y menor prioridad e insertar o eliminar elementos seria mucho mas eficiente. Básicamente una lista simple emporaría mucho la efectividad y complejidad de el manejo de datos por urgencia.

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

Las dos operaciones tienen una complejidad de $O(\log n)$ lo cual hace el heapq sumamente eficiente comparado con otras estructuras de datos similares a la hora de organizar por prioridad.