

Observaciones Laboratorio 9

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

Una cola de prioridad se visualiza como un árbol binario Completo no organizado, sin embargo, para implementar la cola de prioridad se utiliza un array list, ya que con esto es más fácil acceder a los hijos y padres de los nodos.

- 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 y los atiende de esta manera, por eso First In-First Out. Por otra parte, una cola de prioridad organiza los datos según su prioridad, ya no por orden de llegada, y los atiende por 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 en el peor caso se recorre toda la lista [$O(n)$] hasta que en criterio de complejidad se cumpla y volver a recorrerlo hasta que se ordenen todos los datos, en el heap ya se organizan según prioridad y su complejidad es $2n \log n$.

- 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.

Un ejemplo podría ser la asignación de tareas en una empresa según la prioridad en sus operaciones o qué tan rápido tiene que entregarse un proyecto. También se utilizaría una cola de prioridad en la entrada de personas a un vuelo según el pasaje que compraron, donde los que tengan boletos premium se les da mayor prioridad que los de boletos normales.

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

En este caso, se pueden organizar los pacientes en orden de urgencia según su estado de salud, edad o incluso aquellos que pidieron cita con anticipación. Si no es necesaria una cola de prioridad se pueden atender por orden de llegada, ej: Aquellos que no llegaron con cita previa y no tienen alguna urgencia de salud.

Observaciones Laboratorio 9

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

En un heap_pq se accede fácilmente a las posiciones para poder encontrar a los hijos/padres y con ellos comparar para poder manejar prioridades, si se utiliza una lista como single o array donde no se organizan de forma específica (es decir no se implementa el swim/sink) entonces se tendría que recorrer la lista cada vez al momento de ingresar un elemento y tener que acomodarlo según prioridad.

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

Inserción: $O(1 + \log N)$ ya que solo se compara con el padre, se va dividiendo de a dos.

Eliminación: $O(2 * \log N)$ ya que se deben comparar con los dos hijos y se recorre solo la mitad del árbol.