Laboratorio 10 – EDA

Juan Camilo Cancelado – 202410123

Gabriela Gómez - 202420506

Pedro Archila - 202421572

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

- **R:** Array list con la primera posición vacía para poder encontrar fácilmente mediante posiciones los valores con mayor y menor prioridad respectivamente, por ejemplo, en un MaxPQ, si nos ubicamos en cualquier posición y obtenemos su “padre” mediante división entera 2 y, si nos ubicamos en cualquier otra posición obtenemos sus “hijos” mediante 2k y 2k+1

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

- **R:** Un FIFO tradicional se refiere a la funcionalidad de una cola comun y corriente donde el primero que entra es el primero que sale. Por otro lado, en una cola de prioridad el primero de la lista es el primero que sale, sin embargo, esta lista está ordenada por una prioridad, el primero que sale es el que mayor prioridad tiene.

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

- **R:** El módulo que proporciona Python para trabajar mejor con colas de prioridad es el “Heappq” el cual trabaja con un array que tiene la primera posición vacía y este se puede ver como un árbol (sin ser árbol aclaro). Para insertar información se hace uso de “swim” (sube y compara) y para ordenar hace uso de “heap sort” el cual ordena por cada posición y usa “sink” (compara y baja). Cuando se hace heap sort depende del orden se puede hacer maxpq y min pq.

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

- **R:** En las operaciones basicas de heap (buscar, eliminar y agregar), por lo general la complejidad de un array simple es mucho mayor. Por ejemplo, buscar en un array es O(N) pero en heap, como solo recorre una proporcion del array mediante 2k y 2k+1, se ve mucho mas optimizado.

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

- **R:** En este caso primero se atiende el primer elemento que aparece en la lista y después se atiende el otro elemento dado al que tiene menor posición

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

- **R:** Lo fundamental para una cola de prioridad es que la primera posición esté vacía para que se cumpla la propiedad que los hijos de un elemento se encuentran en las posiciones 2k y 2k+1

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

- **R:** La cola de atención en emergencias médicas, Se atienden primero las condiciones más graves y después las menos graves. En la programación de tareas, uno puede organizar el orden en el que se hacen las tareas dependiendo de su complejidad, su importancia, el tiempo que tarda en hacerse, etc.

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

- **R:** La prioridad puede ser tanto la gravedad de la enfermedad como la edad del paciente. Primero se atienden a los pacientes de mayor edad o menor edad o se atienden primero las enfermedades o condiciones más graves.

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

- **R:** Al usar una cola de prioridad, todas las operaciones básicas de añadir valores, eliminar valores y buscar valores se convierten en complejidad de O(Log N). En cambio, utilizando otras estructuras de datos como listas simplemente encadenadas o colas normales la complejidad es mínimo O(N).

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

- **R:** La inserción y la eliminacion de una cola tiene O(logn)