

a) La estructura de datos más usada para implementar una cola de prioridad es un binary heap con un arreglo.

b) Una cola es FIFO, es decir, que atiende a los elementos estrictamente en orden de llegada, sin tener en cuenta la prioridad. Por otro lado, una cola de prioridad siempre atiende el elemento con la prioridad más alta, incluso si se incluyó en la cola después de elementos con menor prioridad.

c) Python proporciona el módulo heapq para tratar este tipo de estructuras fácilmente.

d) Aplicar heaps tiene complejidad logarítmica con constantes de bajo valor para algunas operaciones. Si tenemos una lista ordenada, insertar elementos en la posición correcta es $O(n)$, lo mismo sucede si incluimos al final. Adicionalmente, el heap implementado con arraylist no tiene complejidad de memoria alta.

e) Si dos elementos comparten el mismo nivel de prioridad, el heap solo puede garantizar la propiedad de la propiedad, pero no puede garantizar la estabilidad del orden. En estos casos, se podría atender primero al elemento que llega primero, lo cual fuerza un “desempate”.

f) En un hospital, los pacientes no se atienden de manera FIFO, es decir, si alguien llega al hospital con gripa, no debería ser atendido antes que una persona que llega por una convulsión o por un corte severo, pues claramente es mejor tratar a la persona de más riesgo que a uno que no se vea en peligro si debe esperar quince minutos más. También podríamos pensar en trasplantes de órganos, en donde diversos factores (como la edad, riesgo y causa de la necesidad de trasplante) se usan para calcular la prioridad con la que debería ser puesta una persona en una lista de trasplantes de órganos.

g) En la atención médica, como hemos dicho antes, el riesgo que corre una persona si no es atendida inmediatamente se usa para determinar en qué orden de la lista de prioridad debería ser puesto. Por ejemplo, alguien que sospecha que tiene amigdalitis pero no está seguro en teoría no debería atenderse antes que alguien que acaba de sufrir un accidente de carro.

h) Sustituir un heap por una lista reduce el rendimiento del algoritmo. Esto se debe a que si se inserta algo al final, tenemos $O(1)$ para un heap, pero con la lista si se desea poner algo al final se tiene $O(n)$.

i) En una cola de prioridad con heap: i) inserción: $O(\log(n))$, eliminación $O(\log(n))$, consultar mayor prioridad: $O(1)$.