

Práctica Árboles - EDD

2024-1

Para esta práctica sólo podrán usar árboles como colecciones de datos, específicamente un Heap, los árboles deben estar implementados sobre nodos con enlace a hijo izquierdo y derecho.

Eres un desarrollador de software encargado de crear un sistema de gestión y priorización de pacientes en una sala de urgencias.

Al llegar los pacientes se debe evaluar y asignar el triaje acorde a la siguiente tabla

CLASIFICACION TRIAGE			
NIVEL DE URGENCIA	TIPO DE URGENCIA	COLOR	TIEMPO DE ESPERA
1	RESUCITACION	ROJO	ATENCION DE FORMA INMEDIATA
2	EMERGENCIA	NARANJA	10 - 15 MINUTOS
3	URGENCIA	AMARILLO	60 MINUTOS
4	URGENCIA MENOR	VERDE	2 HORAS
5	SIN URGENCIA	AZUL	4 HORAS

Un paciente con triaje 1, debe ser atendido con prioridad, independientemente del orden de llegada.

Le sigue en prioridad el triaje 2, y así sucesivamente hasta llegar al 5.

Un triaje de número menor siempre tendrá prioridad sobre un triaje de número mayor.

Si se tienen 2 ó más pacientes con el mismo triaje, se debe atender en orden de llegada.

Veamos el siguiente ejemplo, están acorde el orden de llegada y se asigna el correspondiente triaje:

Paciente 1

- Nombre: Pedro
- Edad: 67
- Triage: 4

Paciente 2

- Nombre: Teresa
- Edad: 45
- Triage: 2

Paciente 3

- Nombre: Julio
- Edad: 75
- Triage: 1

Paciente 4

- Nombre: Sofia
- Edad: 15
- Triage: 4

Cuando el doctor esté disponible, el primer paciente a atender es el que llego de 3ra, porque tiene triaje 1

Si no llegan más pacientes, el siguiente paciente a atender es el paciente que llego de 2da, porque tiene triaje 2

Si llegan más pacientes con triaje menor al paciente que llego de primeras estos serán atendidos primero.

En caso contrario, el orden de atención seria Paciente 1 y finalmente Paciente 4 (los dos tienen el mismo triaje, pero el Paciente1 llego primero que el Paciente4)

Requisitos:

Crea una clase llamada Paciente que tenga los siguientes atributos:

Número Paciente (identificador único).

Genero.

Nombre.

Edad

Triage.

La implementación debe crear una cola de prioridad sobre un heap min (utilizando nodos)

El sistema debe permitir realizar las siguientes operaciones

- Registrar (insertar) un paciente, debe ser posible agregar nuevos pacientes, el registro debe conservar el orden de llegada y la prioridad, el triaje 1 debe quedar en la raíz ó seguido de esta dependiendo del orden de llegada.
- Consultar paciente próximo a atención, sin eliminar de la cola de prioridad (solo consulta)
- Opción atender siguiente, en esta opción se debe extraer el paciente que continua en atención acorde a la prioridad y orden de llegada
- Consultar los pacientes que están en espera en general
- Consultar los pacientes que están en espera por triaje
- Opción eliminar paciente, el sistema debe dar la opción si un paciente se desea retirar de la sala de urgencias, este debe ser eliminado, conservando la prioridad de los restantes y el orden de llegada, la eliminación debe ser por nombre y/o por Id.

Entrega y sustentación:

- La fecha de entrega y sustentación es 21 y 22 de mayo (grupo 66 y 65 respectivamente) durante clase.
- Se pueden hacer solos o en parejas, teniendo en cuenta:
 - La sustentación será práctica y se hará de forma individual.
 - La sustentación tendrá una parte oral y es de forma individual.
- La implementación valdrá un 40% y la sustentación un 60%.
 - La nota de implementación depende de la sustentación oral.
- Toda la práctica deberá quedar almacenada en Github y enviada por correo el día de la sustentación plazo máximo 7 am.
- **Si se utiliza una estructura diferente a arboles min heap con nodos, la nota máxima será 1.**