

A thick dark blue vertical bar runs down the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the text 'Algoritmos y Estructuras de Datos'. In the bottom left corner, there are several thin, curved, light blue lines that sweep upwards and to the right.

Algoritmos y Estructuras de
Datos

Trabajo Práctico N° 2

Informe

Ré, Brehm

TECNICATURA UNIVERSITARIA EN PROCESAMIENTO Y
EXPLOTACIÓN DE DATOS



Carrera: Tecnicatura Universitaria en Procesamiento y Explotación de Datos

Cátedra: Algoritmos y Estructuras de Datos

Alumnos: Brehm, Tomás; Ré, Lautaro

Año: 2022

Índice

Informe.....	3
Ejercicio 1: Sala de Emergencias.....	3
Estructura	3
Análisis del orden de complejidad	3
Ejercicio 2: Temperaturas_DB	3
Análisis del orden de complejidad	3
Ejercicio 3: Servicio de Transporte.....	4

Informe

Ejercicio 1: Sala de Emergencias

Estructura

La estructura seleccionada para este ejercicio es un “Montículo binario de mínimos”. La cola de espera se inicializa con éste montículo, cuando un paciente es atendido, se extrae de la estructura.

El primer elemento a extraer es el paciente que tenga el nivel de riesgo más delicado en comparación al resto de pacientes.

El criterio para seleccionar que paciente atender primero cuando ambos poseen el mismo nivel de riesgo, es por turno de llegada, aquel que haya llegado antes, será atendido primero.

Análisis del orden de complejidad

Orden de complejidad para insertar un elemento en el montículo:

- Promedio: $O(\log n)$
- Mejor caso: $O(1)$
- Peor caso: $O(\log n)$

Orden de complejidad para eliminar un elemento en el montículo:

- Promedio: $O(\log n)$
- Mejor caso: $O(1)$
- Peor caso: $O(\log n)$

Ejercicio 2: Temperaturas_DB

Análisis del orden de complejidad

Método	Orden de complejidad	Explicación
trans_fecha(fecha)	$O(1)$	Método implementado para transformar la fecha de formato string a date cada vez que sea necesario.
guardar_temperatura(fecha, temperatura)	$O(\log(N))$	Llama al método “agregar” del AVL.
devolver_temperatura(fecha)	$O(\log(N))$	Utiliza el método “obtener” del AVL para encontrar la clave deseada.
max_temp_rango(fecha1, fecha2)	$O(N \log(N))$	Se itera desde la clave inicial hasta la clave final. Se utiliza el método “obtener” del AVL para recorrer la estructura y encontrar la temperatura máxima en el rango.

min_temp_rango(fecha1, fecha2)	$O(N \log(N))$	Se itera desde la clave inicial hasta la clave final. Se utiliza el método “obtener” del AVL para recorrer la estructura y encontrar la temperatura mínima en el rango.
temp_extremos_rango(fecha1, fecha2)	$O(N \log(N))$	Se llaman a los 2 métodos anteriores, y se obtienen la temperatura máxima y mínima dentro de un rango.
borrar_temperatura(fecha)	$O(\log(N))$	Elimina la temperatura de la base de datos, utilizando el método del AVL “eliminar”.
mostrar_temperaturas(fecha1, fecha2)	$O(N \log(N))$	Se itera desde la clave inicio hasta la final, agregando a una lista la en formato de tuplas, la fecha y temperatura de ese día.
mostrar_cantidad_muestras()	$O(1)$	Retorna el atributo tamaño de la base de datos.

Ejercicio 3: Servicio de Transporte

Para la resolución de este ejercicio, cargamos los datos recibidos en un Grafo. Luego utilizando el algoritmo de dijkstra (Inicializado con una cola de prioridad de máximos), con algunas modificaciones, procesamos los datos del Grafo desde una ciudad origen. De esta manera los atributos de cada vértice se actualizan guardando en la ponderación, el mayor de los mínimos cuello de botella de la capacidad de carga máxima posible para llegar hasta la ciudad destino.

Una vez aplicado el primer algoritmo de dijkstra, se utiliza como filtro el cuello de botella obtenido, para inicializar un segundo Grafo, y así buscar entre las rutas obtenidas, cuál es la menos costosa. Con el Grafo inicializado, aplicamos el algoritmo de dijkstra (Inicializado con una cola de prioridad de mínimos), para obtener la ruta con el precio total más bajo.

Por último, recorreremos desde el Vértice final, hacia el Vértice inicial, en el último Grafo procesado, por medio de los predecesores, cada ciudad y vamos agregándolas a una lista para tener la ruta final. Una ruta para llegar desde la ciudad inicial hasta la ciudad destino, con la mayor capacidad de carga y el menor precio total posible.