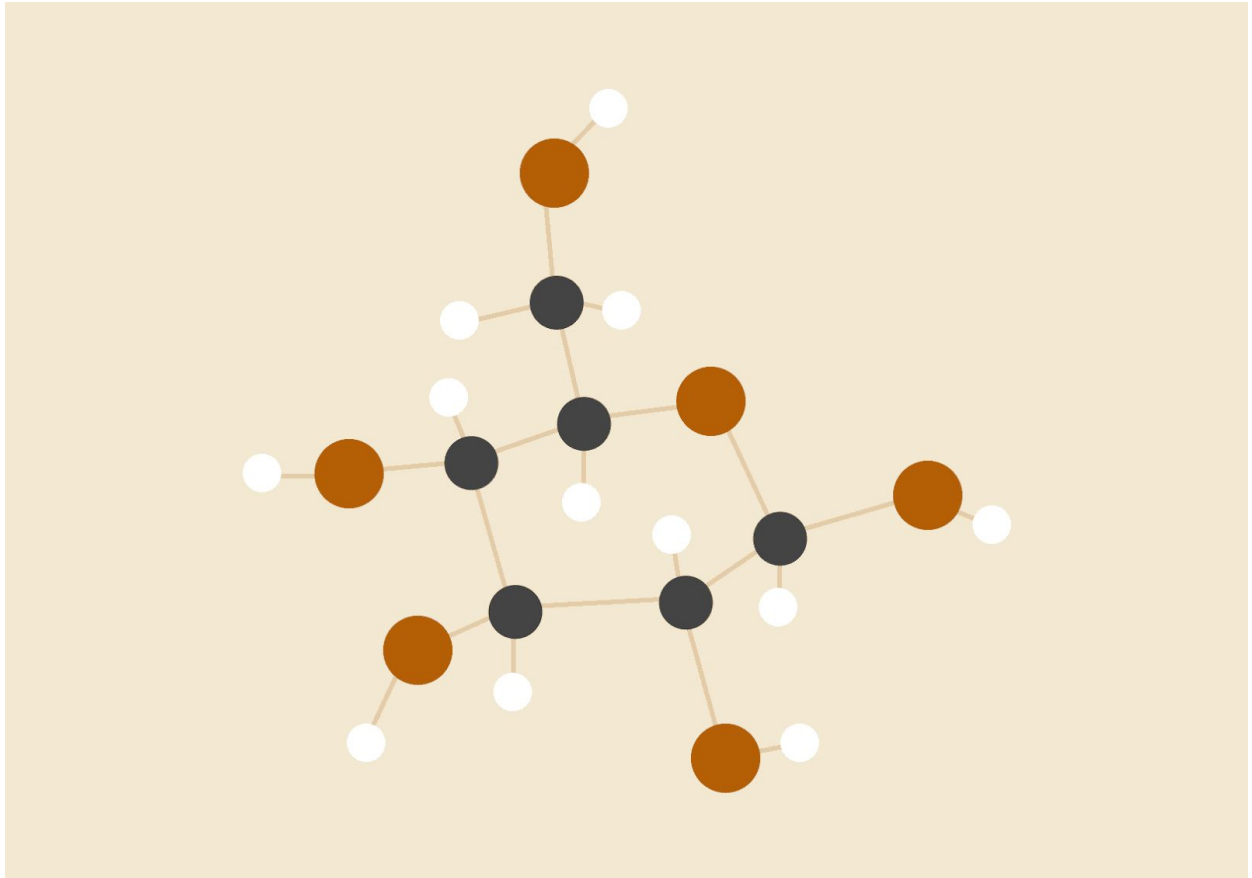


Informe Clinica Zyxcba



Federico Ezequiel Jaleh. Padron: 105553

Francisco Orquera Lorda. Padron 105554

Corrector: Lautaro Manzano

Informe

- Primero se nos pasan por parámetro dos archivos csv, uno con doctores y su especialidad, y otro con los pacientes, y sus años de inscripción.
- Elegimos un hash para el guardado de doctores y pacientes, debido a que las búsquedas del mismo son en $O(1)$ y no necesitábamos un orden para los mismos. Para el hash de doctores, la clave es el nombre del doctor y el dato es la especialidad del mismo. Para los pacientes, la clave es el nombre del paciente, y el dato del año de ingreso.
- Ambos hashes, utilizan la función free para destruir el dato, debido a que, leyendo línea por línea, estamos liberando los datos de ambos hashes.
- Luego, creamos un hash para las especialidades de los doctores con clave el nombre de la especialidad y dato un TDA que modela una lista de espera, desarrollado en los próximos ítems.
- También creamos un ABB para los doctores que iban atendiendo a cada paciente, con clave el doctor, y dato el paciente atendido. La decisión del uso del ABB fue debido a que a la hora de pedir el informe de la clínica, en el caso de ser un rango acotado de doctores, la búsqueda debía ser en tiempo logarítmico.

TDA Espera

- Elegimos crear nuestro propio TDA ya que necesitábamos poder manejar cierta prioridad en la espera de la clínica. La consigna nos decía que había tanto pacientes urgentes, como regulares y los urgentes tenía más prioridad a la hora de ser atendidos por los doctores.
- Por lo tanto, para poder manejar esta prioridad a la hora de atenderlos, decidimos crear un TDA Espera, el cual consta de otros dos TDAs, una Cola y un ABB.
- Para guardar a los pacientes urgentes, utilizamos una Cola ya que el orden relativo de la misma es del tipo FIFO (first in, first out). Esto es importante ya que los turnos urgentes se atienden por orden de llegada. Además, cumplía con la condición del enunciado, la cual decía que atender un paciente urgente sea en $O(1)$.

- Para guardar a los pacientes regulares utilizamos un ABB que tiene como clave el año de ingreso del paciente, y como dato una Cola en la que guardamos los nombres de los pacientes. La decisión de usar un ABB fue porque el enunciado pedía que en el caso de pacientes regulares debíamos atender primero a los que lleven más tiempo en la clínica. Este dato era sencillo de encontrar utilizando un ABB y además cumplía con el tiempo pedido ($O(\log(p))$), siendo “p” cantidad de pacientes en esa especialidad).
Dentro del ABB utilizamos una Cola ya que podemos tener claves repetidas (mismo año de inscripción). Si dos pacientes se registran en el mismo año, ambos son guardados en la Cola y son desencolados a la hora de ser atendidos, respetando así el orden de inscripción en la clínica.

Modificación del ABB

- Tuvimos que modificar la primitiva `abb_iter_in_crear` ya que para pedir el informe de los doctores el enunciado nos aclaraba que se nos podía pasar un rango de doctores. En el caso de que se especifique un inicio de rango, primero se llamará a una función para encontrar ese inicio o en su defecto el posterior más cercano. Llenamos el iterador por rangos, donde el primer elemento que se itera es el elemento encontrado cuando buscamos el inicio, de esta manera cumplimos, por cómo busca el `abb`, con el tiempo pedido por el enunciado para rangos cortos ($O(\log(d))$), siendo “d” la cantidad de doctores en el sistema).