


A dark blue vertical bar is on the left. A blue arrow points right from it, containing the date.

2-12-2021

Trabajo Practico 2

Clinica zyxcba

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

Integrantes: Galarza Adrián (106194), Scazzola Martin (106403)
Facultad de Ingeniería - Universidad de Buenos Aires
Materia: Algoritmos y Programación II
Ayudante Asignado: Martin Buchwald

Introduccion:

Se buscará con los conocimientos adquiridos a lo largo de la materia modelar el Sistema de funcionamiento de una clínica en lenguaje C, utilizando todos los conocimientos incorporados sobre este lenguaje de programación y los TDA necesarios, vistos a lo largo del cuatrimestre.

Funcionamiento general de la clínica:

El sistema inicia recibiendo por comandos un archivo de pacientes, con su respectivo año de inscripción y un archivo de doctores, con su respectiva especialidad, este los lee y carga en memoria, almacenándolos en estructuras de datos acorde a las necesidades utilitarias y de complejidad temporal. Con los datos cargados en memoria el programa opera por entrada estándar, donde se pueden ejecutar los siguientes comandos:

Pedir turno:

Se recibe el nombre del paciente y el nombre de la especialidad, y el sistema lo añade a la lista de espera correspondiente.

Atender siguiente paciente:

Se recibe el nombre del doctor que quedo libre y este atiende al siguiente paciente urgente (por orden de llegada), si no hubiese pacientes urgentes, atiende al siguiente paciente con mayor antigüedad en la clínica.

Informe doctores:

Se imprime la lista de doctores en orden alfabético, junto con su especialidad y el número de pacientes que atendieron desde que arrancó el sistema.

Opcionalmente, se puede especificar el rango (alfabético) de doctores sobre los que se desean informes.

Llegado al caso que alguno de los comandos no cumpla con las especificaciones el sistema automáticamente imprime por salida estándar el error correspondiente y continua con el programa, leyendo la siguiente línea de la entrada estándar.

El sistema finaliza una vez leída por completo la entrada estándar, cerrando todos los archivos leídos y liberando toda la memoria utilizada.

Estructuras de datos y TDA utilizados:

Para poder implementar el sistema de la clínica, se tuvieron que utilizar diferentes tipos abstractos de datos vistos a lo largo del cuatrimestre, estos nos facilitan diferentes tareas y nos permiten almacenar datos y operar con ellos, cumpliendo con la complejidad temporal pedida para los comandos.

Los pacientes son cargados en un hash para cuando se pide un turno, poder comprobar si este es socio de la clínica en tiempo constante.

Las diferentes especialidades se almacenaron en un hash, donde la clave era el nombre de la especialidad y el dato era un nuevo TDA específico que se implementó, llamado lista de espera, que internamente tenía un atributo que era una cola para los pacientes urgentes, para poder atenderlos en tiempo constante y un heap para los regulares sabiendo que nos podíamos relajar a atenderlos en tiempo logarítmico por orden respecto a la antigüedad en la clínica.

Para almacenar los datos de los doctores se decidió utilizar un árbol binario de búsqueda, debido a que este TDA garantiza la propiedad de poder recorrerlo en orden, lo cual es una ventaja bastante útil sabiendo que se debe implementar un comando para obtener el informe de los doctores de la clínica en orden alfabético, además obtener dicho informe tendrá una complejidad temporal logarítmica en caso de que se especifique un rango y en el peor que se quiera un informe de todos los doctores lineal.

Utilizando estas estructuras de datos, el comando atender siguiente cumple con la complejidad pedida, debido a que, si se tratase de un paciente urgente, habría que comprobar que el doctor pertenezca a la clínica, lo cual sería $O(\log d)$ siendo d la cantidad de doctores cargados en el sistema, posteriormente buscar en el hash de especialidades la especialidad correspondiente al doctor, $O(1)$ y luego desencolar de la cola de urgentes de dicha especialidad, $O(1)$. Para los casos regulares sería el mismo procedimiento, pero habría que desencolar del heap de regulares, lo cual sería de complejidad $O(\log n)$, siendo n la cantidad de pacientes regulares en la lista de espera. En conclusión, para los casos urgentes el comando atender siguiente es de complejidad $O(\log d)$, y para los casos regulares $O(\log d + \log n)$.

