



Tecnológico de Monterrey

Programación de estructuras de datos y algoritmos fundamentales

**Act 1.3 - Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales
(Evidencia Competencia)**

Javier Corona Del Río. A01023063

Kevin López Cano A01028138

Preguntas:

¿Cuántos registros tiene tu archivo?

R=Nuestro archivo tiene 36673 registros.

Para esta pregunta se leyó el archivo csv con un while, que a su vez, gracias al ADT, asignó cada variable del archivo a los parámetros del objeto Reader creando entonces un objeto por línea del archivo, estos luego se agregaron a un vector de objetos. Se imprimió el size con la función size() de la librería vector para saber cuántos objetos se crearon, es decir, cuantos registros había en nuestro .csv y poder responder la pregunta.

¿Cuántos récords hay del segundo día registrado? ¿Qué día es este?

R=En el segundo día de registros, es decir (11-08-2020) se hicieron 3,305 registros.

Debido a que los registros ya estaban ordenados por fecha no se ordenó el archivo; procedemos directamente a la búsqueda secuencial para encontrar la fecha 11-08-2020 y que se sumara un contador cada vez que la encontrara, la búsqueda se detenía si la fecha en la cual estaba comparando ya era mayor a la fecha que estábamos buscando, por ello este método de búsqueda fue el más eficiente para esta pregunta.

¿Alguna de las computadoras pertenece a Jeffrey, Betty, Katherine, Scott, Benjamin, Samuel o Raymond?

R=Si, alguna de las computadoras si pertenece a al menos alguno de ellos.

Para esta pregunta primero organizamos el vector en orden alfabético tomando únicamente la variable hostmail_origin y posteriormente para optimizar el código procedimos a la búsqueda de benjamin.reto.com por el método secuencial ya que este empieza por b entonces en caso de estar en el archivo ya se detendría la búsqueda. Primero usamos selection sort pero se tardaba aproximadamente 2 a 3 minutos porque se comparaba cada variable con las demás, debido a eso optamos por implementar Quicksort, este fue más rápido y redujo considerablemente el tiempo de ordenamiento de esta pregunta.

Por otra parte cabe resaltar que si no se ordenan los servidores por orden alfabético benjamin aparece en una posición mucho más cercana al inicio por lo que se podría considerar si organizarlo vale la pena o no.

¿Cuál es la dirección de la red interna de la compañía?

R= La dirección de la red interna de la compañía es “10.221.242.0”.

Para responder esta pregunta se usó un while, el cual evaluaba si una variable string era igual a “-”, es decir, siempre y cuando la variable red fuera vacía el while seguiría buscando, pero si encontraba una red, que por cierto todas eran iguales, dejaría de buscar y extraería la información hasta antes del tercer punto es decir así: “10.221.242” , luego le agregamos un “.0”, este equivale al número de computadora de cada usuario de la red, en definitiva quedo así: “10.221.242.0”.

¿Alguna computadora se llama server.reto.com?

R= No, no hay ninguna computadora llamada “server.reto.com”.

Aprovechando que el vector ya estaba ordenado por orden alfabético en la variable servidor, se utilizó binary search, esta decisión la tomamos ya que “server. reto.com” se encontraba en el final del vector (empieza por s) entonces la búsqueda binaria sería más eficiente que la secuencial.

Debido a que la búsqueda binaria devuelvio la posición -1 llegamos a la conclusión de que “server.reto.com” no estaba.

¿Qué servicio de mail utilizan (algunas ideas: Gmail, Hotmail, Outlook, Protonmail)?

R= El servicio de mail que utiliza la compañía es freemailserver.com

Para esta pregunta lo que hicimos fue buscar mediante búsqueda secuencial el número de veces que se repetía la dirección IP de la compañía, está estaba 13381 veces y la dirección de mail asociada a esta IP era “freemailserver.com” por lo que llegamos a la conclusión de que este era el servicio de mail que utilizaba la compañía.

Considerando solamente los puertos destino ¿Qué puertos abajo del 1000 se están usando? Lista los puertos e investiga qué aplicación/servicio lo utiliza generalmente.

Los puertos debajo de 1000 son:

993: Este es el puerto que debes usar si quieres conectarte usando IMAP de forma segura (cifrada). Internet Message Access Protocol (IMAP) es un protocolo de correo usado para acceder al correo de un servidor web remoto desde un cliente local.

465: Este es el puerto que debes usar si quieres conectarte usando SMTP de forma segura (cifrada). Simple Mail Transfer Protocol (SMTP) es un protocolo estándar para el envío de correo de Internet.

443: Este puerto es también para la navegación web, pero en este caso usa el protocolo HTTPS que es seguro y utiliza el protocolo TLS por debajo.

80: Este puerto es el que se usa para la navegación web de forma no segura HTTP.

¿Qué hizo cada quien?

ReadLines.h:

Aquí definimos nuestro ADT creando una clase llamada “Reader” que se encargaría de crear un objeto por registro y cada uno de ellos con sus respectivas variables, la implementación de la clase la hizo Kevin pero los getters que luego fueron utilizados para acceder a las variables del objeto en las funciones de comparación los implemento Javier.

SortSearch.h:

En este archivo fue donde creamos nuestras clases de búsqueda y ordenamiento utilizando templates, el desarrollo de este archivo fue muy equitativo ya que nos dividimos la implementación de dichas clases, Javier al inicio había implementado el selectionSort pero como dijimos anteriormente, este se estaba tardando mucho en ordenar por lo que Kevin luego implementó el quickSort, por otro lado la búsqueda secuencial de la fecha y la implementación de la búsqueda binaria para el “server.reto.com” y Kevin implemento la búsqueda secuencial para saber si alguna de las computadoras pertenecía a Jeffrey, Betty, Katherine, Scott, Benjamin, Samuel o Raymond y también implementó la búsqueda secuencial del IP destino.

main.cpp:

En este archivo, cada uno se encargó de implementar las funciones de comparación para las respectivas búsquedas y ordenamientos que describimos en el SortSearch.h, pero Kevin se encargó de implementar el while que permite la lectura del archivo y la creación de los objetos en el vector, finalmente el pequeño while que anteriormente describimos que utilizamos para la red interna de la compañía lo implementamos los dos con ayuda de algunos blocks de programación en internet, porque este requiere de utilizar dos funciones de la librería string y ninguno de los dos estaba muy seguro de cómo hacerlo.

Reflexión individual:

Esta actividad tarea considero estuvo a la altura y no decepcionó. Abarcó la mayoría de los temas que vimos y las diferentes implementaciones. En varias partes terminamos cambiando las cosas porque era muy complejo lo que queríamos implementar y no salía o porque era más eficiente de otra manera cómo fue el caso del selection sort. Logré comprender más a fondo los temas de la clase, cómo su implementación y el trabajar cómo leer cada unidad de la base de datos. Me gusto mucho la actividad por todo lo que me aportó en términos de conocimiento.