Tecnológico de Monterrey

Campus:

Santa Fe

Materia:

Programación de estructuras de datos y algoritmos fundamentales

Nombre de la actividad:

Act 3.2 - Actividad Integral de BST (Evidencia Competencia)

Nombre y Matrícula:

Emiliano Deyta Illescas A01785881
Darío Cuauhtémoc Peña Mariano A01785420

Profesor:

Vicente Cubells

Grupo:

601

Fecha:

01/11/2024

1. ¿Existe algún sitio que se mantenga en el top 5 todos los días?

No, ninguno se mantuvo en el top 5 todos los días, cambiaron la mayoría todos los días

```
------------------------------------------------------------
Top 5 conexiones para la fecha 10-8-2020:
netflix.com con 16 conexiones
usatoday.com con 16 conexiones
microsoft.com con 16 conexiones
etsy.com con 15 conexiones
go.com con 14 conexiones
------------------------------------------------------------
Top 5 conexiones para la fecha 11-8-2020:
netflix.com con 15 conexiones
instagram.com con 15 conexiones
tripadvisor.com con 15 conexiones
steamcommunity.com con 14 conexiones
bestbuy.com con 14 conexiones
------------------------------------------------------------
Top 5 conexiones para la fecha 12-8-2020:
mail.yahoo.com con 18 conexiones
etsy.com con 17 conexiones
businessinsider.com con 16 conexiones
live.com con 15 conexiones
medicalnewstoday.com con 15 conexiones
------------------------------------------------------------
Top 5 conexiones para la fecha 13-8-2020:
medicalnewstoday.com con 18 conexiones
weather.com con 16 conexiones
cnn.com con 16 conexiones
instagram.com con 15 conexiones
chase.com con 15 conexiones
------------------------------------------------------------
Top 5 conexiones para la fecha 14-8-2020:
healthline.com con 19 conexiones
foodnetwork.com con 17 conexiones
accuweather.com con 17 conexiones
craigslist.org con 16 conexiones
medicalnewstoday.com con 16 conexiones
------------------------------------------------------------
Top 5 conexiones para la fecha 17-8-2020:
hulu.com con 16 conexiones
usnews.com con 15 conexiones
live.com con 15 conexiones
fandom.com con 14 conexiones
webmd.com con 14 conexiones
```

```
------------------------------------------------------------
Top 5 conexiones para la fecha 18-8-2020:
indeed.com con 19 conexiones
quizlet.com con 16 conexiones
lowes.com con 16 conexiones
apple.com con 16 conexiones
google.com con 15 conexiones
------------------------------------------------------------
Top 5 conexiones para la fecha 19-8-2020:
xfinity.com con 19 conexiones
homedepot.com con 17 conexiones
steampowered.com con 15 conexiones
cnet.com con 15 conexiones
etsy.com con 14 conexiones
------------------------------------------------------------
Top 5 conexiones para la fecha 20-8-2020:
homedepot.com con 518 conexiones
expedia.com con 21 conexiones
fandom.com con 18 conexiones
indeed.com con 17 conexiones
dictionary.com con 16 conexiones
------------------------------------------------------------
Top 5 conexiones para la fecha 21-8-2020:
quizlet.com con 17 conexiones
target.com con 17 conexiones
youtube.com con 16 conexiones
bbb.org con 16 conexiones
yellowpages.com con 16 conexiones
```

Código:

```cpp
template <typename T>
void BSTLectura<T>::top(const std::vector<T>& db, const std::string& fecha, int topN)
{
    std::map<std::string, int> conexionesMap = conexionesPorDia(db, fecha);
    std::vector<conexionesEntrantes<T> > conexionesVec;
    for (typename std::map<std::string, int>::iterator it = conexionesMap.begin(); it != conexionesMap.end(); ++it)
    {
        conexionesVec.push_back(conexionesEntrantes<T>(it->first, it->second));
    }
    std::sort(conexionesVec.begin(), conexionesVec.end());
    std::cout << "------------------------------------------------------------\n";
    std::cout << "Top " << topN << " conexiones para la fecha " << fecha << ":\n";
    for (int i = 0; i < topN && i < conexionesVec.size(); ++i)
    {
        std::cout << conexionesVec[i] << std::endl;
    }
}
```

2. ¿Existe algún sitio que entre al top 5 a partir de un día y de ahí aparezca en todos los días subsecuentes?

No, todos cambian cada fecha, ninguno se repite.

```
------------------------------------------------------------
quizlet.com aparece en 21-8-2020 y en todos los días subsecuentes.
------------------------------------------------------------
target.com aparece en 21-8-2020 y en todos los días subsecuentes.
------------------------------------------------------------
youtube.com aparece en 21-8-2020 y en todos los días subsecuentes.
------------------------------------------------------------
bbb.org aparece en 21-8-2020 y en todos los días subsecuentes.
------------------------------------------------------------
yellowpages.com aparece en 21-8-2020 y en todos los días subsecuentes.
------------------------------------------------------------
```

Código:

```cpp
template <typename T>
void BSTLectura<T>::apareceUnDiaYSubsecuentes(const std::map<std::string, std::vector<conexionesEntrantes<T> > >& topnporfecha)
{
    std::set<std::string> dominiosImpresos;
    for (typename std::map<std::string, std::vector<conexionesEntrantes<T> > >::const_iterator it = topnporfecha.begin(); it != topnporfecha.end(); ++it)
    {
        for (typename std::vector<conexionesEntrantes<T> >::const_iterator vecIt = it->second.begin(); vecIt != it->second.end(); ++vecIt)
        {
            if (dominiosImpresos.find(vecIt->dominio) != dominiosImpresos.end()) continue;

            bool apareceEnTodos = true;
            for (typename std::map<std::string, std::vector<conexionesEntrantes<T> > >::const_iterator sigIt = std::next(it); sigIt != topnporfecha.end(); ++sigIt)
            {
                if (!dominioEnConexiones(sigIt->second, vecIt->dominio))
                {
                    apareceEnTodos = false;
                    break;
                }
            }

            if (apareceEnTodos)
            {
                std::cout << "------------------------------------------------------------\n";
                std::cout << vecIt->dominio << " aparece en " << it->first << " y en todos los días subsecuentes.\n";
                dominiosImpresos.insert(vecIt->dominio);
            }
        }
    }
}
```

3. ¿Existe algún sitio que aparezca en el top 5 con una cantidad más alta de trafico que lo normal?

Si, Home depot aparece en el día 20-8-2020 con 518 conexiones

```
------------------------------------------------------------------------
Sitios con conexiones superiores al promedio para el día 20-8-2020:
homedepot.com con 518 conexiones
------------------------------------------------------------------------
```

Código:

```cpp
template <typename T>
void BSTLectura<T>::sitioConMuchasConexiones(const std::map<std::string, std::vector<conexionesEntrantes<T> > >& topnporfecha)
{
    for (typename std::map<std::string, std::vector<conexionesEntrantes<T> > >::const_iterator it = topnporfecha.begin(); it != topnporfecha.end(); ++it)
    {
        int totalConexiones = 0;
        for (typename std::vector<conexionesEntrantes<T> >::const_iterator vecIt = it->second.begin(); vecIt != it->second.end(); ++vecIt)
        {
            totalConexiones += vecIt->cantidad;
        }
        double promedio = static_cast<double>(totalConexiones) / it->second.size();
        std::cout << "------------------------------------------------------------\n";
        std::cout << "Sitios con conexiones superiores al promedio para el día " << it->first << ":\n";
        for (typename std::vector<conexionesEntrantes<T> >::const_iterator vecIt = it->second.begin(); vecIt != it->second.end(); ++vecIt)
        {
            if (vecIt->cantidad > promedio)
            {
                std::cout << vecIt->dominio << " con " << vecIt->cantidad << " conexiones\n";
            }
        }
    }
}
```

Reflexión Individual (Emiliano Deyta A01785881):

El uso de un Árbol Binario de Búsqueda (BST) es una opción eficiente para el monitoreo y análisis de redes debido a su estructura ordenada, que permite realizar búsquedas, inserciones y eliminaciones en un tiempo logarítmico promedio, O(log n). En redes donde se gestionan grandes volúmenes de conexiones, un BST permite organizar y acceder a los datos de manera rápida y ordenada, lo cual es esencial para identificar patrones o anomalías en el tráfico de red en tiempo real.

En el contexto de detección de infecciones, un BST facilita la búsqueda y clasificación de conexiones que podrían indicar actividad sospechosa. Al filtrar y organizar las conexiones por criterios específicos, como la dirección IP o el puerto, el BST permite identificar patrones de comportamiento inusuales, como un número anormal de conexiones hacia destinos desconocidos o repeticiones frecuentes de conexiones en ciertos puertos (como el 80 y el 443). Este tipo de análisis es clave para identificar posibles infecciones, pues permite detectar picos de tráfico o comportamientos anómalos que pueden sugerir la presencia de malware o actividad maliciosa.

Para determinar si una red está comprometida, el BST se puede combinar con algoritmos de detección de patrones y machine learning que analicen los datos filtrados. Esto permite que el BST funcione como un sistema de prefiltrado, identificando rápidamente las conexiones más relevantes y permitiendo que otras herramientas se encarguen de un análisis profundo.

Reflexión Individual (Darío Peña A01785420):

La gestión de grandes volúmenes de datos en redes es un desafío constante, y elegir la estructura de datos adecuada puede marcar la diferencia en términos de eficiencia y efectividad. Un Árbol Binario de Búsqueda (BST) es una de las opciones para organizar y acceder a estos datos de manera ordenada. En el contexto de monitoreo de redes, un BST permite filtrar y categorizar conexiones por criterios como la dirección IP o el puerto de destino, lo que es útil para detectar patrones de tráfico y posibles amenazas.

La principal ventaja de un BST es su capacidad para realizar operaciones de búsqueda e inserción de forma rápida, lo que facilita identificar y analizar conexiones que podrían ser preocupantes. Por ejemplo, si se desea observar únicamente las conexiones hacia puertos como el 80 y el 443, un BST puede ayudar a filtrar estos registros, permitiendo enfocarse en aquellos que podrían ser un indicio de comportamiento anómalo o actividad maliciosa. De esta manera, se reduce la carga de análisis al priorizar los datos más relevantes.

Sin embargo, aunque un BST es útil, no es una solución definitiva. Un árbol puede volverse ineficiente si no está bien equilibrado, lo que podría afectar el tiempo de búsqueda y organización de datos. En redes más complejas, otras estructuras o combinaciones de métodos podrían ser más adecuadas. No obstante, un BST sigue siendo una herramienta valiosa para tener un primer filtro y estructura de los datos de conexión, permitiendo que otras herramientas de análisis profundicen en los detalles y características más complejas del tráfico de red.

En conclusión, aunque un BST no garantiza la detección directa de infecciones en una red, sí proporciona un método ordenado y eficiente para manejar grandes volúmenes de datos y detectar patrones básicos que podrían sugerir la necesidad de un análisis más exhaustivo. Como parte de un enfoque más amplio de monitoreo y seguridad, un BST puede ser el primer paso para entender mejor el tráfico de una red y protegerla de posibles amenazas.