PUCE Virtual

DOCUMENTO DE PROFUNDIZACIÓN

Programación III

Laboratorio de Contenido 2: Árboles Balanceados – Problema del Desbalance

Laboratorio de Contenido 2: Árboles Balanceados – Problema del Desbalance

Explicación básica

Un árbol balanceado mantiene su altura controlada para garantizar operaciones en O(log n).

Objetivos

- Comprender el impacto del desbalance.
- Comparar rendimiento entre árbol balanceado y degenerado.

Caso práctico

Un sistema almacena IPs sospechosas en orden creciente:
[10, 20, 30, 40, 50]

Problema

Insertar estos valores en un BST normal.

Aplicación paso a paso

1. Insertar 10 → raíz.
2. Insertar 20 → derecha.
3. Insertar 30 → derecha de 20.
4. Insertar 40 → derecha de 30.
5. Insertar 50 → derecha de 40.

Resultado: árbol degenerado (como lista).
Búsqueda pasa de O(log n) a O(n).

Implementación en Python

```
valores = [10, 20, 30, 40, 50]

bst = BST()
for v in valores:
   bst.raiz = bst.insertar(bst.raiz, v)

print("Árbol construido (estructura degenerada)")
```

Conclusiones

Cuando los datos llegan ordenados, el BST pierde eficiencia. En ciberseguridad, esto puede afectar búsquedas masivas de logs.