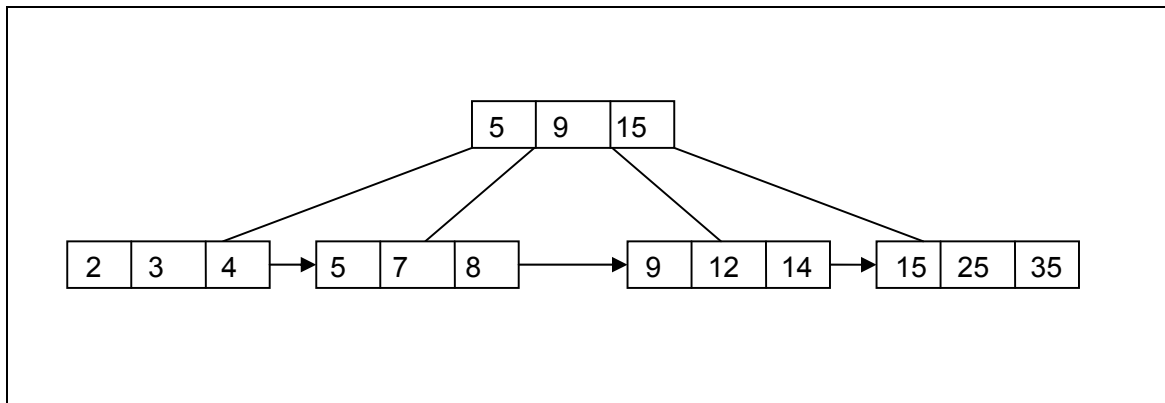


Estructuras de Datos II

Segundo Parcial

Los árboles B+ son estructuras de datos jerárquicas que se utilizan para almacenar y manipular datos ordenados de forma muy eficiente, ya que por su estructura y sus propiedades las inserciones y las eliminaciones se realizan en un tiempo logarítmico amortizado. Por esta razón, se utilizan para crear índices dentro de las bases de datos relacionales, para agilizar la búsqueda de registros.

Un nodo de un árbol B+ de orden n , al igual que en un árbol B, puede tener (n) hijos, por lo cual contendrá $n-1$ claves. Por ejemplo, un árbol B+ de orden 4 tiene la siguiente estructura:



En esta estructura (para este caso, de orden 4), se debe garantizar que:

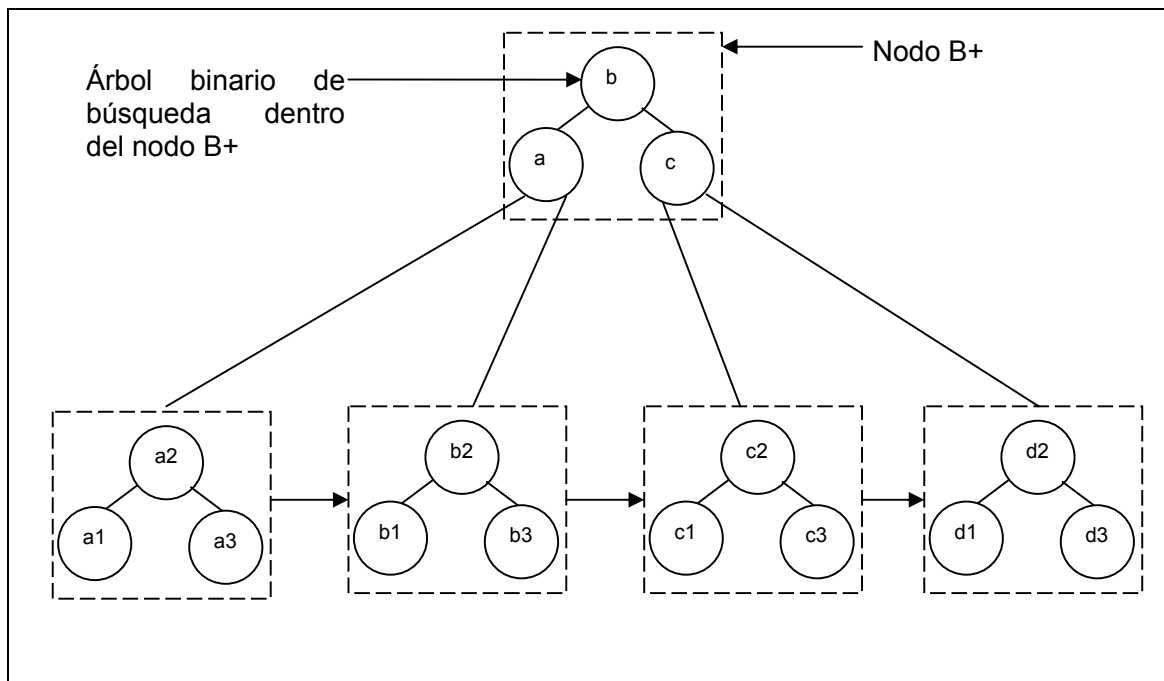
- Los datos dentro de cada nodo se encuentran ordenados de forma ascendente.
- Todos datos de los hijos almacenados a la izquierda de cualquier dato deben ser menores que él y todos los datos de los hijos almacenados a la derecha de cualquier dato deben ser mayores o iguales que él.

Igualmente se deben cumplir las siguientes propiedades:

- Todas las hojas del árbol se encuentran en el mismo nivel
- Los nodos de un mismo nivel se encuentran unidos en una lista enlazada simple

Debido a que cada nodo puede contener varios datos, es común implementarlos como árboles binarios de búsqueda con $n-1$ nodos (si el árbol B+ es de orden n , el árbol binario de búsqueda de cada nodo deberá contener $n-1$ nodos binarios). Esto evita la sobrecarga de implementar los nodos del árbol B+ como arreglos, además que ahora espacio al no tener que almacenar arreglos con algunas posiciones vacías.

Por ejemplo, para un árbol B+ de orden 4, la estructura de los nodos será la siguiente:



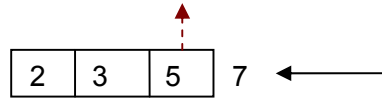
Inserción en un árbol B+

El proceso de insertar un valor en un árbol B+ se realiza **siempre en los nodos hoja**, de la siguiente forma:

- Si el árbol está vacío, se debe crear un nuevo nodo B+ e insertar el nuevo valor (Esto incluye crear el árbol binario dentro del nodo B+ e insertar el valor). Este nodo se convierte en la raíz del árbol.
- Si el árbol no está vacío, primero se deberá verificar que el nuevo dato no exista dentro del árbol. Para ello se deberá realizar una búsqueda empezando desde la raíz. Si el nuevo dato no existe, el proceso de búsqueda deberá hallar el nodo hoja (B+) en el cual se va a insertar el nuevo nodo.
- Una vez encontrado el nodo hoja (B+) en el cual se almacenará el nuevo dato se pueden presentar dos casos:
 - La hoja (B+) tiene espacio para almacenar el nuevo dato. Se debe insertar el dato dentro del nodo B+ y terminar.
 - La hoja (B+) no tiene espacio: En este caso la hoja se deberá dividir en dos nodos B+. Se deberán disponer los valores (que estaban almacenados en la hoja más el nuevo valor) de forma ascendente y se tomará el valor de la mitad. Este valor se deberá llevar al nodo B+ padre (si no existe, se deberá crear uno). El hijo izquierdo del dato que se llevó al padre deberá contener los valores menores que él en la hoja original, y el hijo derecho deberá contener **el dato que se llevó al padre** y los datos mayores que él en la hoja original.
- Si al llevar un dato al nodo padre no existe espacio para almacenarlo, se deberá dividir el nodo en dos (como en el paso descrito anteriormente), pero **el dato que se lleva al nodo padre no deberá ser almacenado el hijo derecho**.

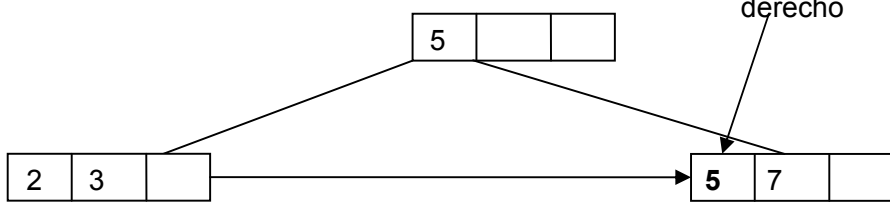
A continuación se ilustra gráficamente el proceso de insertar datos dentro de un árbol B+.

Insertar 2, 3, 5, 7



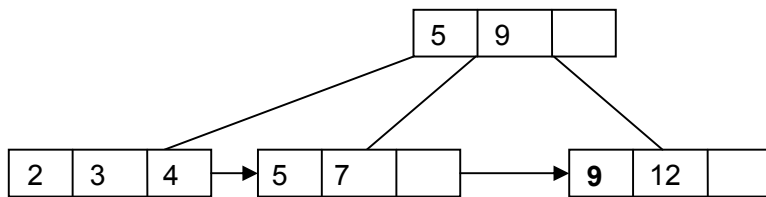
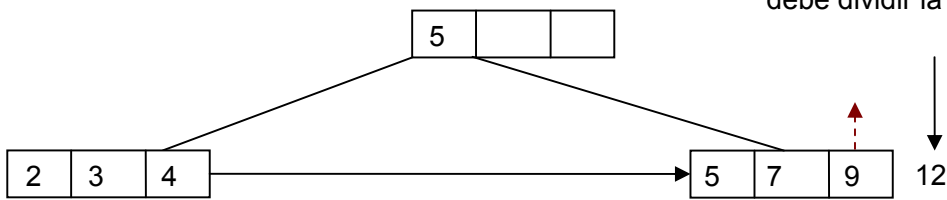
No hay espacio, se debe dividir la hoja.

Cuando se divide una hoja, se conserva el dato que se subió al padre en su hijo derecho



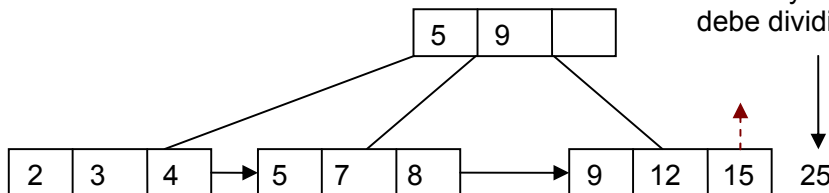
Insertar 4, 9, 12

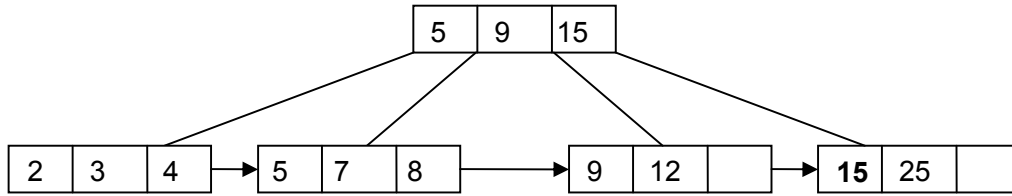
No hay espacio, se debe dividir la hoja.



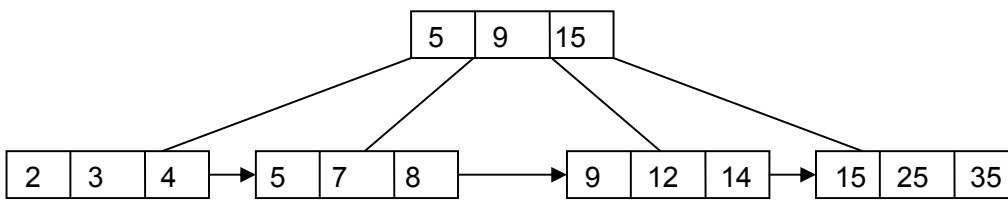
Insertar 8, 15, 25

No hay espacio, se debe dividir la hoja.



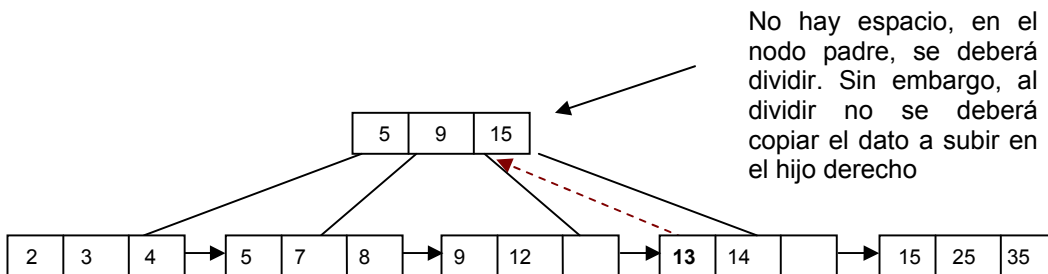


Insertar 14, 35, 13

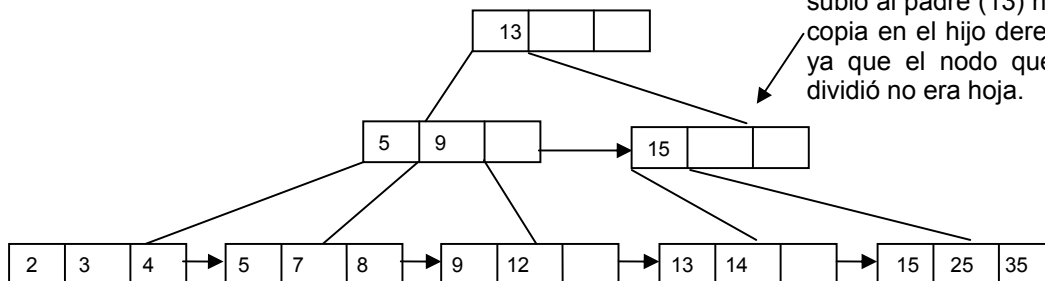


13

No hay espacio, se debe dividir la hoja.



No hay espacio, en el nodo padre, se deberá dividir. Sin embargo, al dividir no se deberá copiar el dato a subir en el hijo derecho



Observe que en este caso el dato que se subió al padre (13) no se copia en el hijo derecho, ya que el nodo que se dividió no era hoja.

Problema

Se deberá implementar un programa lea comandos desde la entrada estándar (que podrá ser redireccionada desde un archivo) para insertar y buscar datos en un árbol B+ que se encuentra inicialmente está vacío. El programa deberá terminar cuando encuentre una línea en el archivo con la palabra "SALIR".

Suponga que se va a utilizar el árbol B+ para almacenar las llaves primarias de una tabla en una base de datos relacional, con el fin de optimizar su proceso de búsqueda. Cada vez que se realiza una inserción, la nueva llave deberá ser insertada en el árbol B+, y cada vez que se busque una llave dentro del árbol se deberá imprimir el número de pasos que se realizaron para encontrar la llave, comparados con el número de pasos necesarios para buscar en una lista simple enlazada.

A continuación se muestra un ejemplo del funcionamiento del programa.

Ejecución	Archivo ejemplo.txt
bplus.exe < ejemplo.txt	INSERTAR 2
	INSERTAR 3
	INSERTAR 5
Pasos para buscar 2:	INSERTAR 7
B+: 3 Lista: 1	INSERTAR 4
Pasos para buscar 35:	INSERTAR 9
B+: 5 Lista: 12	INSERTAR 12
Pasos para buscar 13:	INSERTAR 8
B+: 3 Lista: 13	INSERTAR 15
	INSERTAR 25
	INSERTAR 14
	INSERTAR 35
Fin de la ejecución.	INSERTAR 13
	BUSCAR 2
	BUSCAR 35
	BUSCAR 13
	SALIR

Se debe tener en cuenta que es posible encontrar el valor que se está buscando en un nodo que no sea hoja (por ejemplo, al buscar el número 13 en el árbol de ejemplo se puede encontrar en un solo paso, ya que se encuentra en la raíz del árbol), pero se desea obtener la respuesta del peor caso, es decir, cuando el dato a buscar se encuentra en un nodo hoja del árbol B+ (En el ejemplo, para encontrar el 13 hay que realizar 3 pasos).

Los comandos que se deben implementar son:

INSERTAR <i>valor</i>	Inserta <i>valor</i> dentro del árbol B
BUSCAR <i>valor</i>	Busca <i>valor</i> dentro del árbol B+ e imprime el número de pasos que tuvo que realizar para encontrar el valor. En caso que no se encuentre el valor en el árbol también deberá imprimir el número de pasos que realizó. El número de pasos debe incluir la

	búsqueda dentro de cada nodo B+.
SALIR	Termina la ejecución.