

Arboles

Dado el siguiente árbol B+ de orden 5. La raíz se debe mantener en la posición 0 del archivo. Política de Derechos. Realizar las operaciones: $+5\%$, $+5\%$, -8% , -8% . Indicar el costo de cada operación (Lecturas y Escrituras).

Nodo 0: 4, 1, 1 (23)2 (52) 3 (73) 4 (88) 5

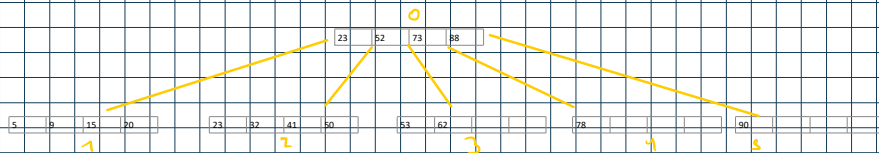
Nodo 1: 4, 1, 1 (5) (8) (15) (20) \rightarrow 2

Nodo 2: 3, 1, 1 (23) (32) (41) (50) \rightarrow 3

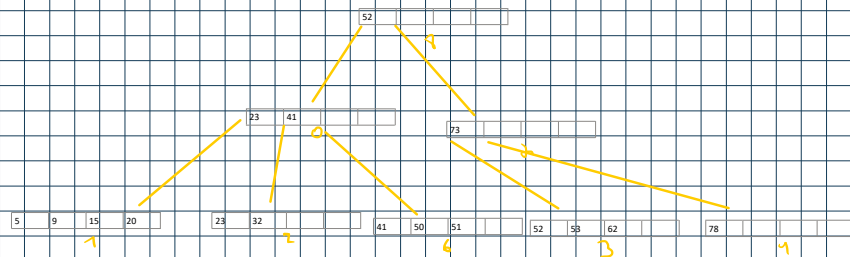
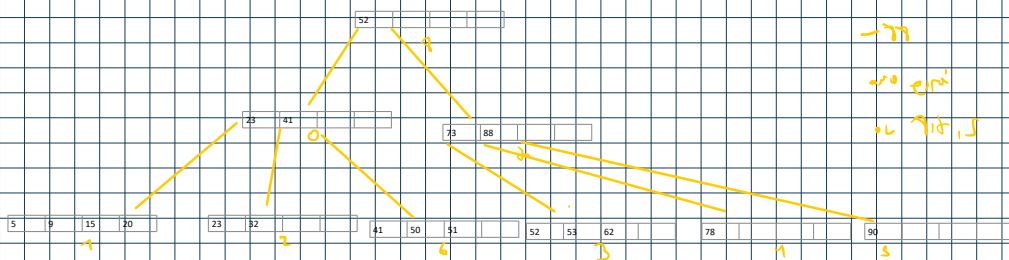
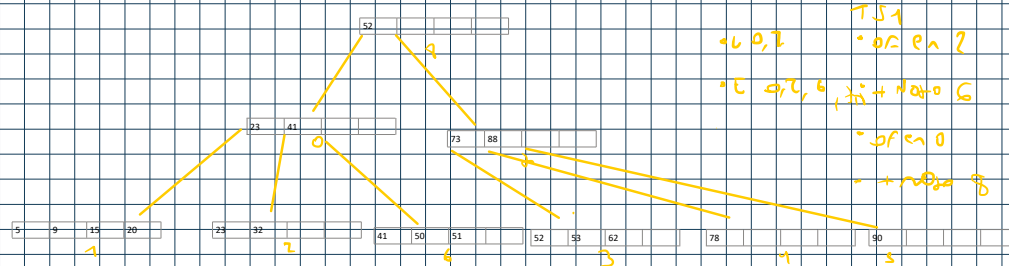
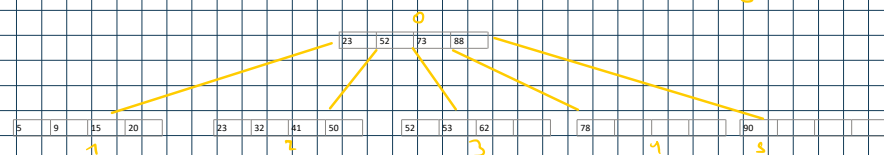
Nodo 3: 2, 1, 1 (53) (62) \rightarrow 4

Nodo 4: 1, 1, 1 (78) \rightarrow 5

Nodo 5: 1, 1, 1 (90) \rightarrow 1



+5%
+5%
-8%

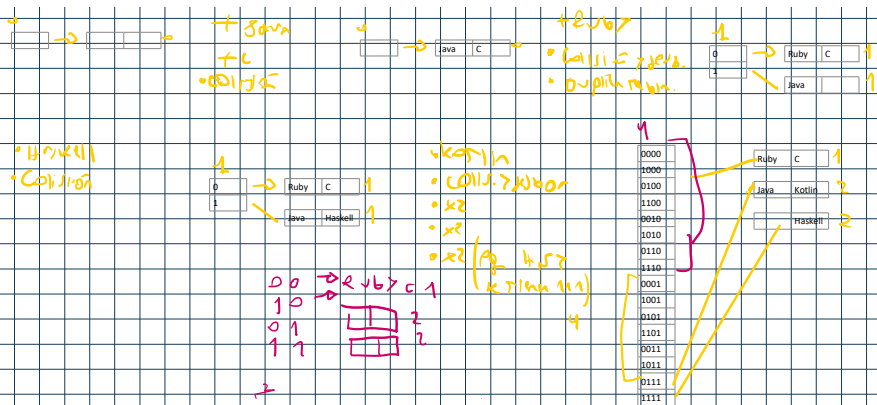


-751
0.2 en 2
0.2 en 2, 6, 10 + 10 = 0.6
0.2 en 0
+ 100 = 0.8

Hashing

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las claves. Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondientes. Justifique brevemente.

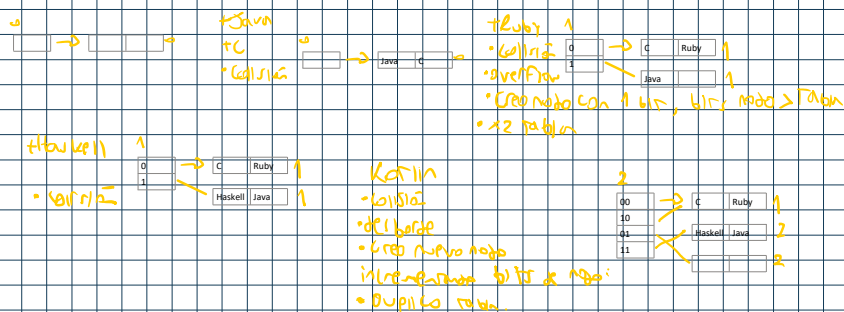
| | | | | | |
|---|--------|---------|---|---------|---------|
| 1 | Java | 0100111 | 2 | C | 0101010 |
| 3 | Ruby | 1011110 | 4 | Haskell | 1001111 |
| 5 | Kotlin | 1010111 | 6 | Python | 1110000 |
| 7 | PHP | 1011101 | 8 | SQL | 0011011 |



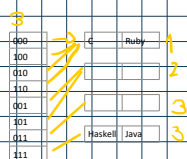
Hashing

Realice el proceso de dispersión mediante el método de hashing extendible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las claves. Deberá explicar los pasos que realice en cada operación y dibujar los estados sucesivos correspondientes. Justifique brevemente.

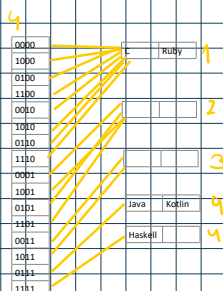
| | | | | | |
|---|--------|---------|---|---------|---------|
| 1 | Java | 0100111 | 2 | C | 0101010 |
| 3 | Ruby | 1011110 | 4 | Haskell | 1001111 |
| 5 | Kotlin | 1010111 | 6 | Python | 1110000 |
| 7 | PHP | 1011101 | 8 | SQL | 0011011 |



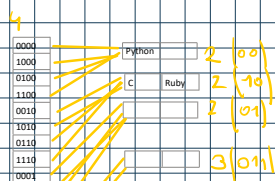
- Siguen llegando de nuevo, nuevo + bits para poder direccionar a Haskell, Java > Kotlin.
- Creo otra nota incrementando su cont de bits.
- bits nota > bits de tabla, duplico tabla.



- Vgo en la situación anterior, creo un nuevo nodo incrementando cont de bits. Duplico tam. tabla.
- Ahora es fácil direccionar los elementos. Acomodo entres.

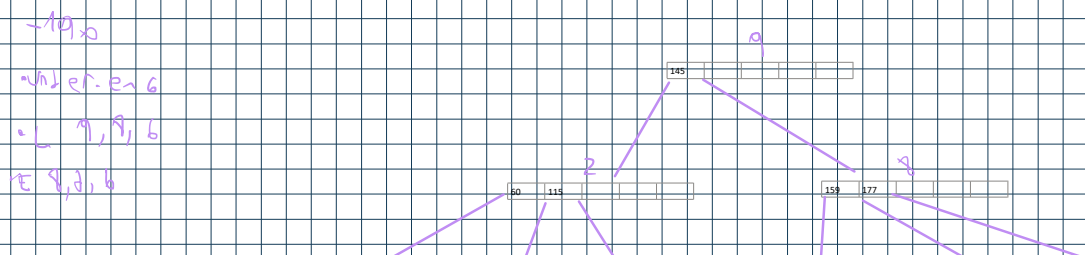
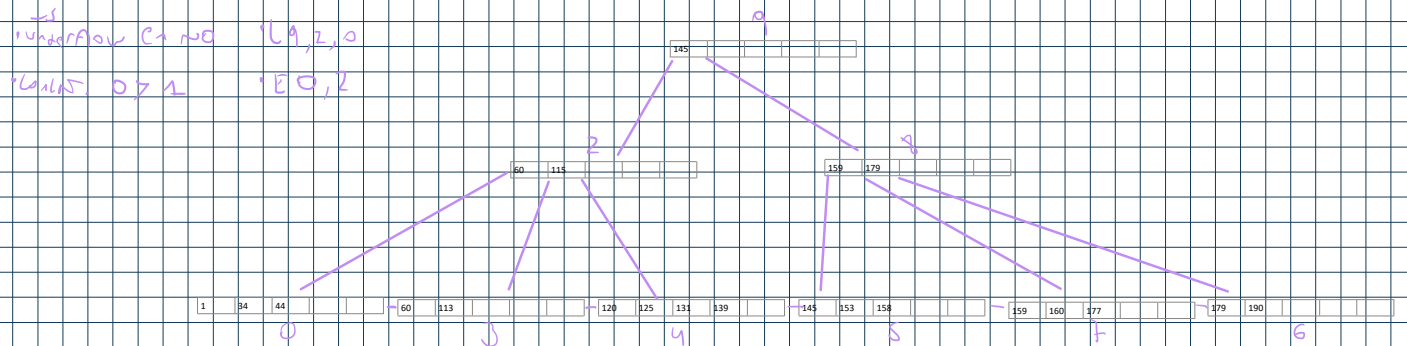
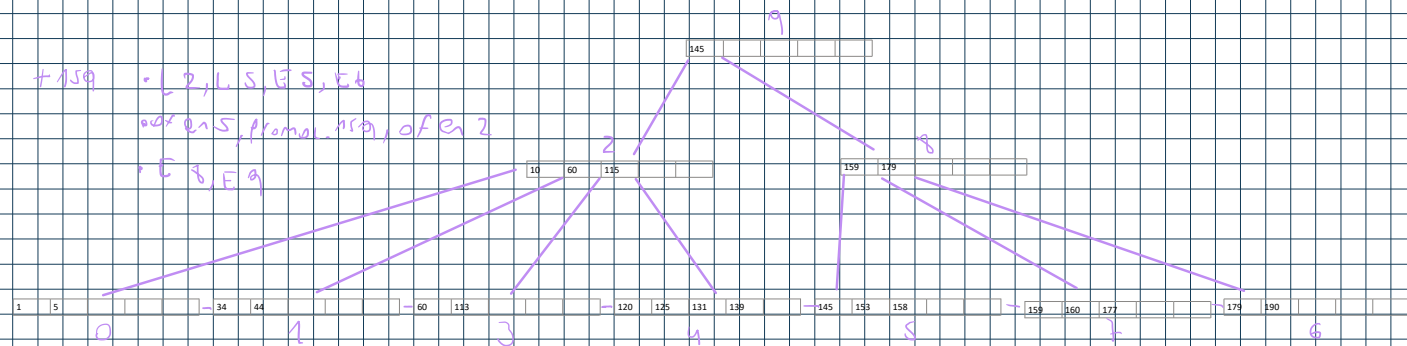
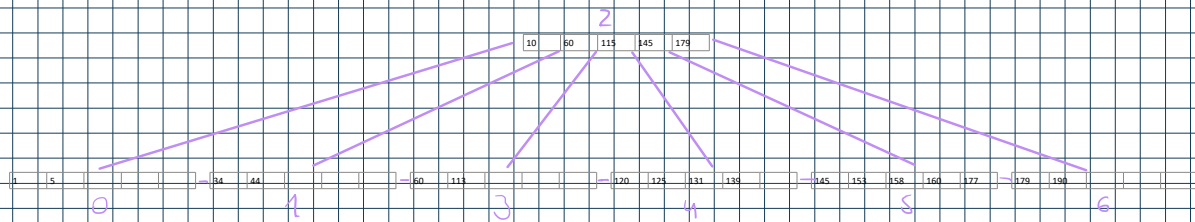
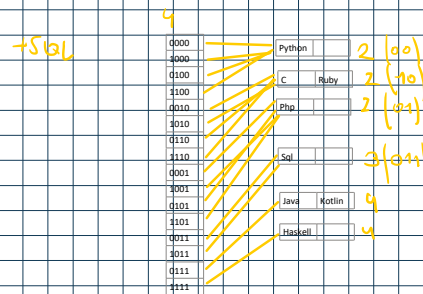
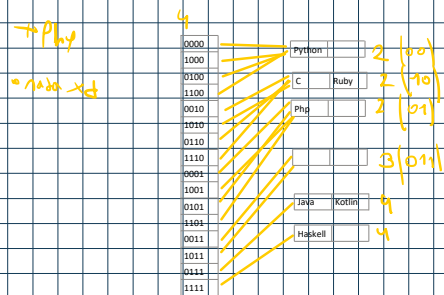


- + Python
- nuevo + de nuevo
- nuevo nodo.



Hand-drawn diagram of a 12-bit memory stack. The stack grows downwards from 0011 to 1111. It contains values 2, 3, 4, and 4 at various addresses. Languages Java, Kotlin, and Haskell are also indicated.

| Address | Value | Language |
|---------|-------|--------------|
| 1100 | | |
| 0011 | 2 | |
| 1010 | | |
| 0110 | | |
| 1110 | 3 | |
| 0001 | | |
| 1001 | | |
| 0101 | 4 | Java, Kotlin |
| 1101 | | |
| 0011 | 4 | Haskell |
| 1011 | | |
| 0111 | | |
| 1111 | | |



E 8, 9, 6

