

Parcial 1

Arboles

$$\max = n - 1 = 4$$

$$\min = \lceil n/2 \rceil - 1 = 1$$

B+ Detallecha

O: 1(23) 2(52) 3(73) 4(88) 5

1:(5)(9)(15)(20) → 2:(23)(32)(41)(50) → 3:(53)(62) → 4:(78) → 5:(90) → -1

[+52] Va en el nodo 3, entra Costo Op: L0 L3 E3

O: 1(23) 2(52) 3(73) 4(88) 5

1:(5)(9)(15)(20) → 2:(23)(32)(41)(50) → 3:(52)(53)(62) → 4:(78) → 5:(90) → -1

[+51] Va en el nodo 2 y genera overflow, el overflow genera overflow en el nodo 0 también Costo Op: L0 L2 E2 E6 E7 E8 E0

$$OF(2): \underbrace{23 - 32}_{n_2} - \underbrace{\overbrace{41}^{\text{promoción}} - 50 - 51}_{n_6}$$

$$OF(0): \underbrace{23 - 41}_{n_7} - \underbrace{\overbrace{52}^{n_0} - 73 - 88}_{n_8}$$

O: 7(52) 8

7: 1(23) 2(41) 6

8: 3(73) 4(88) 5

1:(5)(9)(15)(20) → 2:(23)(32) → 6:(41)(50)(51) → 3:(52)(53)(62) → 4:(78) → 5:(90) → -1

[-88] Costo Op: L0 L8 L5 (no encuentra)

[-90] Genera underflow en 5, como no tengo hermano derecho intento reagrupar con el izquierdo, como no puedo fusionar y eliminar elementos separados 78 88 90

O: 7(52) 8

Costo Op: L0 L8 L5 L4 E4 E8 -5 11

7: 1(23) 2(41) 6

8: 3(73) 4(88) 5

1:(5)(9)(15)(20) → 2:(23)(32) → 6:(41)(50)(51) → 3:(52)(53)(62) → 4:(78)(88) → -1

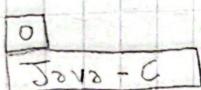
Hashing

① x ②

Tabla de memoria

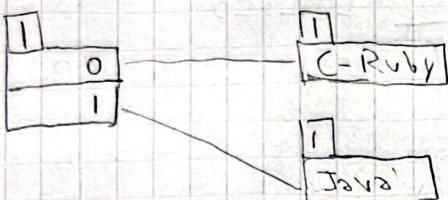


Nodos en disco

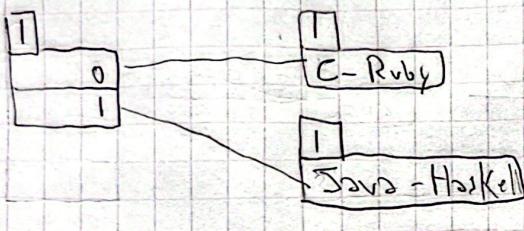


Para insertar Java y C
tengo lugar

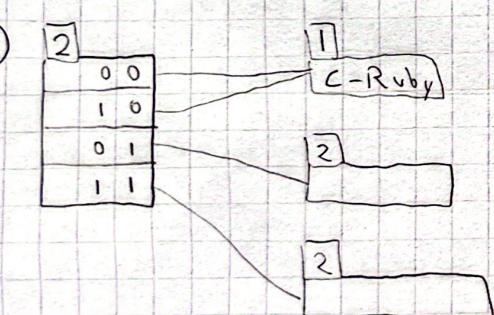
③



④



⑤



Cada nodo tiene espacio para
2 elementos. En la tabla de memoria

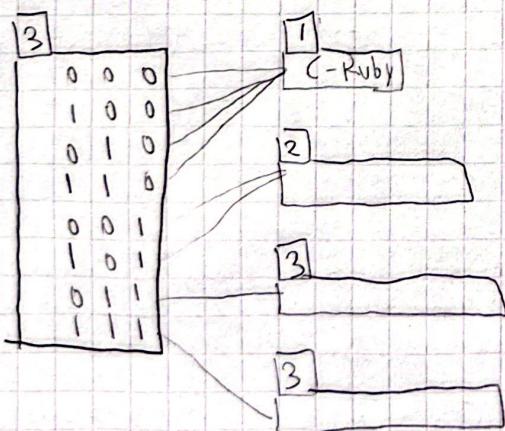
el numero m significa que
hay 2^m direcciones. En los

nodos en disco n representan
ceros de los últimos dígitos
de la dirección de memoria
tendremos en cuenta para
almacenar un elemento allí.

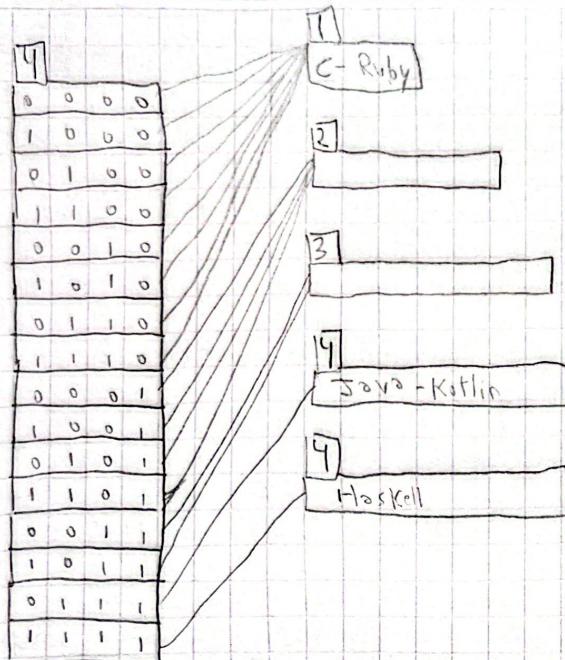
m siempre debe ser igual

al mayor n de los nodos en
disco. Si hay saturación al insertar en
nodo en disco, aumentar el valor asignado
 $n >$ ese nodo y crear otro nodo
igual. si $n > m$ incremento m
y duplico direcciones

Java - Haskell - Kotlin no entran, overflow



Java - Haskell - Kotlin no entran, overflow



⑥, ⑦, ⑧ entran todos por lo que los inserto en orden Secuencial

