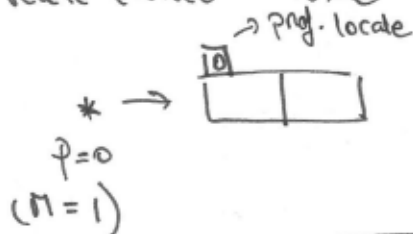


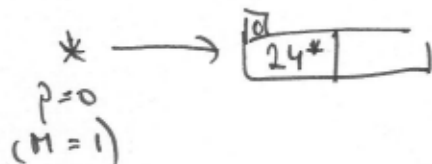
1. Début; hash index "vide"

- profondeur globale (p) = 0 ; $M = 2^0 = 1$
- une seule bucket (vide), avec une profondeur locale = 0
- une seule entrée "vide" dans le pointer directory

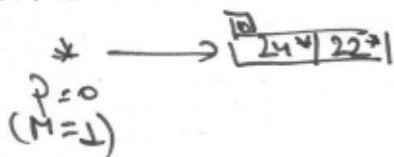


2. Insertion de 24^* (= entrée de données avec clé = 24)
 $h(24) = 01000101$

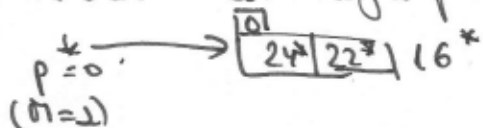
Comme $p=0$ nous examinons les " p dernier bits" → donc en réalité rien du tout.
 Nous insérons simplement dans la (seule et unique) bucket disponible.



3. Insertion de 22^* : même principe.

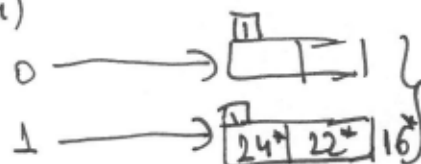


4. Insertion de 16^* même principe, mais la bucket est déjà pleine!



Besoin de split; comme la profondeur locale de la bucket est égale à la profondeur globale (ou, autrement dit, la bucket est "pointée par une seule entrée du pointer directory"), nous devons doubler le directory (et donc augmenter p)

(suite 4)



les deux buckets résultant du split;
 leur profondeur locale est de 1 (= la profondeur de la bucket d'origine + 1)

$p=1$
 $(M=2^p=2)$
 a augmenté de 1

Comme $p=1$ maintenant, nous devons considérer 1 bit pour distribuer les entrées de la bucket d'origine dans les deux buckets résultantes.

$$h(24) = 01000101$$

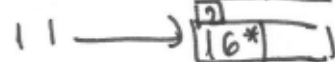
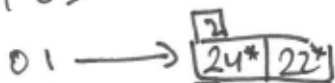
$$h(22) = 11110001$$

$$h(16) = 10101011$$

Malheureusement, les trois entrées tombent dans la même bucket résultante (celle correspondant à 1). Il y a donc toujours dépassement de capacité et nous devons refaire un split, cette fois-ci de la bucket pointée par 1. Comme sa profondeur locale est égale à la profondeur globale, nous devons doubler le pointer directory (et donc augmenter p).

(suite 4)

L'index devient



$p=2$
 $(M=2^p=4)$

profondeur locale ++
car ce sont des buckets résultant d'un split.

La bucket "vide" garde sa profondeur locale à 1 (elle n'a pas été concernée par le split) et elle est maintenant pointée par deux entrées : 00 et 10 (ce sont les deux entrées "résultant de l'entrée 0" après doublement du dictionnaire).

On redistribue les entrées de la bucket splitée dans les deux buckets résultantes, en prenant en compte 2 bits (car $p=2$)

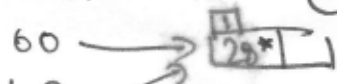
$b(24) = 01000101$

$b(22) = 11110001$

$b(16) = 10101011$

5. Insertion de 28*

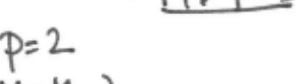
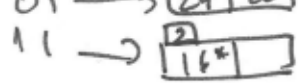
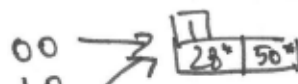
$h(28) = 00000110$



$p=2$
 $(M=4)$

6. Insertion de 50*

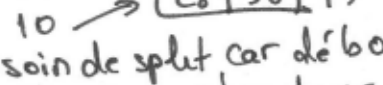
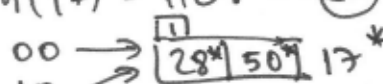
$h(50) = 11110010$



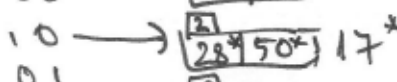
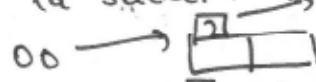
$p=2$
 $(M=4)$

7. Insertion de 17*

$h(17) = 11010110$

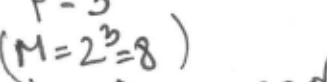
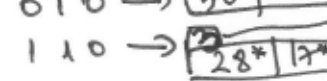
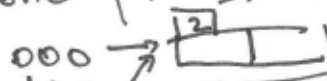


Besoin de split car débordement. Comme la profondeur locale de la bucket est inférieure à la profondeur globale ($1 < 2$), nous pouvons effectuer un split sans doublement; nous "réutilisons" une des entrées pointant sur la bucket : prof. locale augmentée car split



n'a pas augmenté
 $p=2$
 $(M=4)$

Malheureusement, même après split, il y a toujours débordement (sur la bucket pointée par 10); il faut refaire un split, cette fois-ci avec doublement (donc $p++$) → reste la même



$p=3$
 $(M=2^3=8)$

$h(28) = 00000110$

$h(50) = 11110010$

$h(17) = 11010110$