



# TD<sub>2</sub>

## exercice 1 :

on va stocker par rapport au titre

si ordre = 2

combien d'enregistrement on peut avoir par page ?

↳ au maximum  $2 \times \text{Ordre} = 4$

ce veut dire qu'il ya  $\text{ord} \leq \text{nbr d'enregistrements} \leq \text{ord}$

$$2 \leq k \leq 4$$

$\uparrow$                              $\uparrow$   
min                            max

on commence par le racine :

le SGBD dans les métadonnées il a l'adresse

de la racine (le 1<sup>er</sup> noeud), c'est ce dont il a besoin pour trouver la table

les métadonnées

table Film = Nil

nom des attributs

taille des attribut

ordre 2

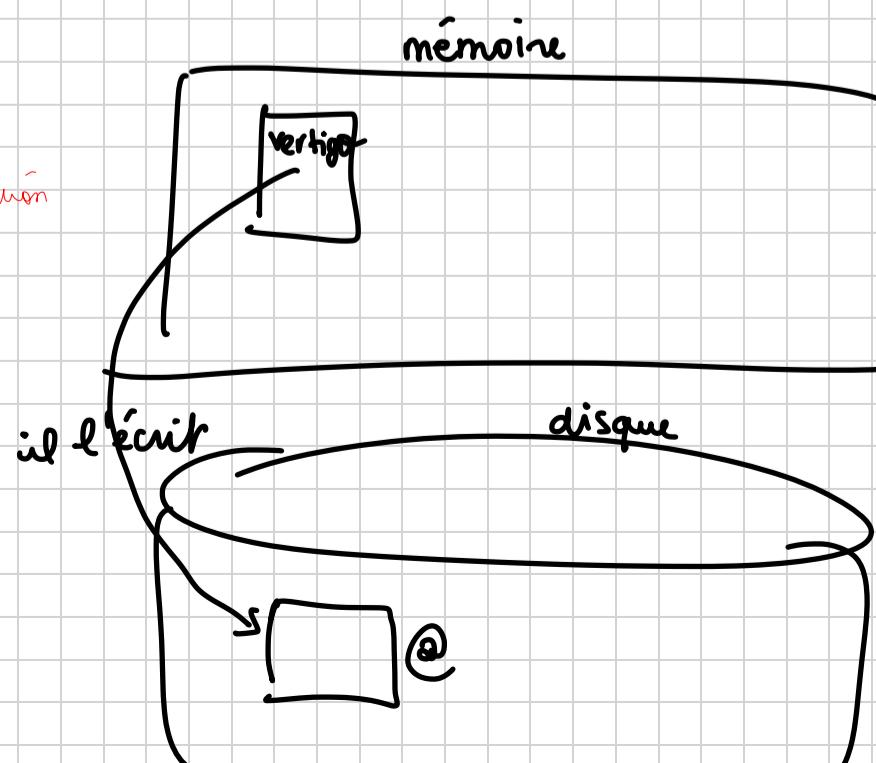
Arbre B

on a pas  
encore  
la page

↓  
on exécute l'insertion  
- stocke vert

vertigo

arrive



## les métadonnées

→ et donc  
ça devient

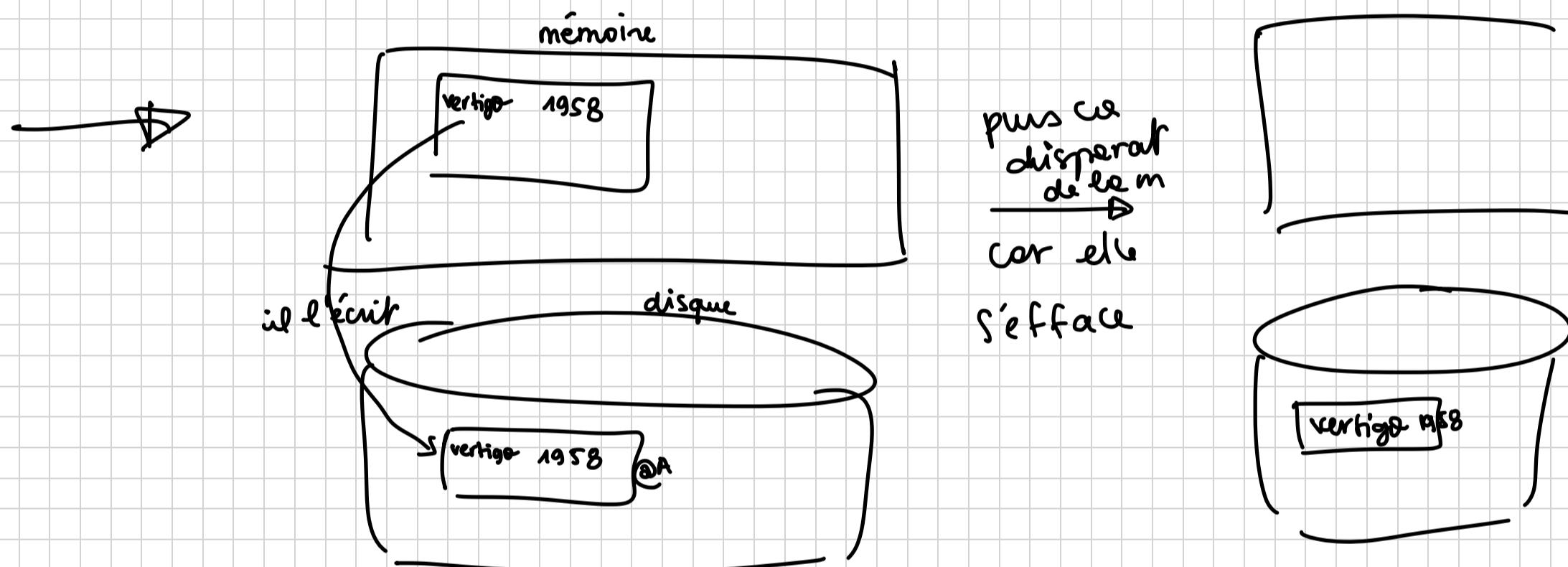
table Film = @ A

nom des attributs

taille des attribut

ordre 2

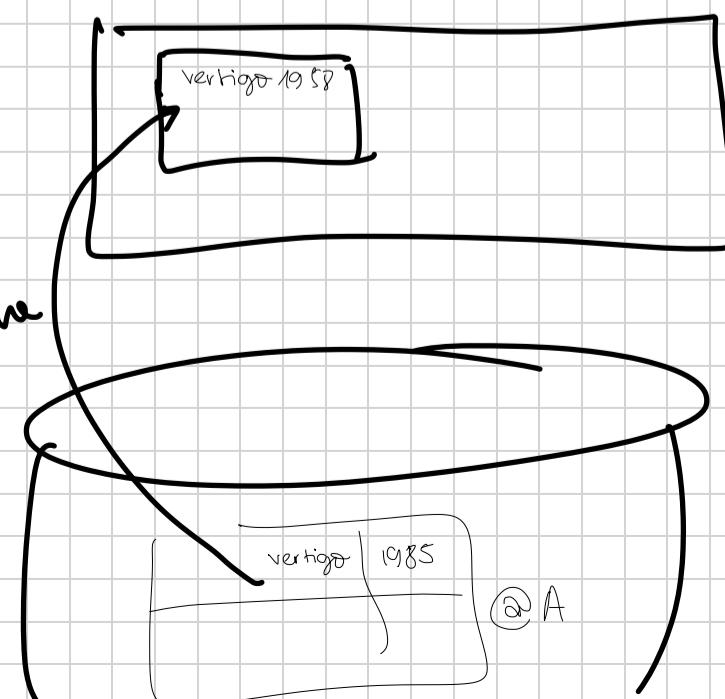
Arbre B



il cherche le movie  
et le charge en memoire  
(lit)

brésil arrive  
(2jrs après) dans  
les données

l'instruction c'est  
l'insert brésil 1984

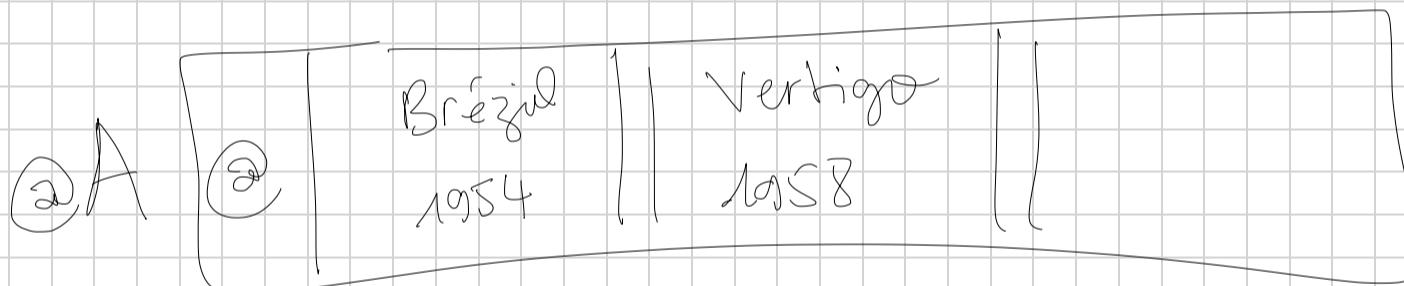




il décole et insère  
car c'est dans ordre  
lexicographique

car on est en train  
de stocker par  
rapport à l'attribut titre

on lit le page disque  
puis on ajoute dans le mémoire  
puis on l'écrit sur le disque.

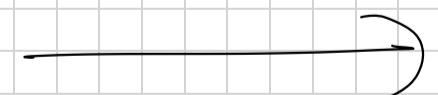


on insère Twin  
peaks



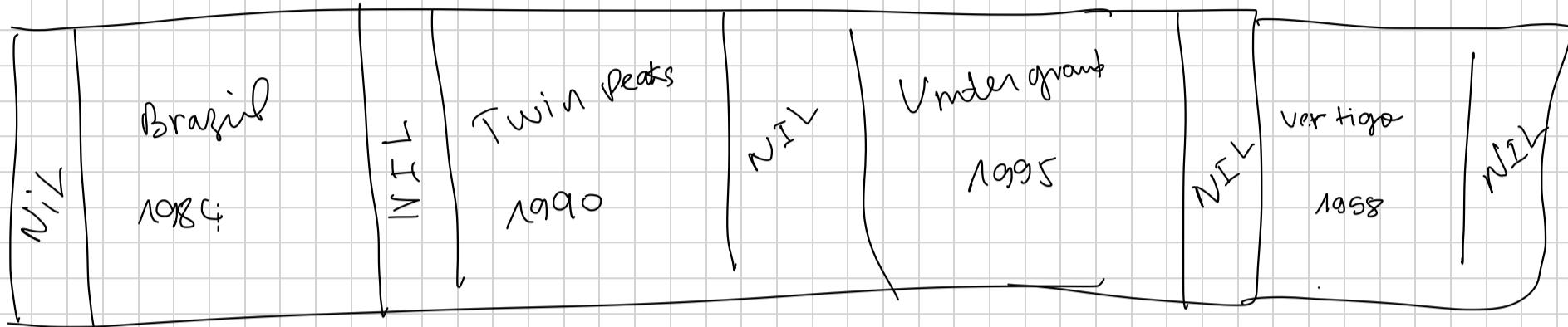
on insère le

route au final on a  $S^a$

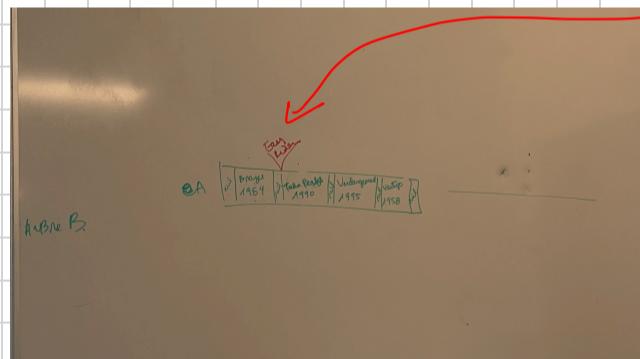




② A



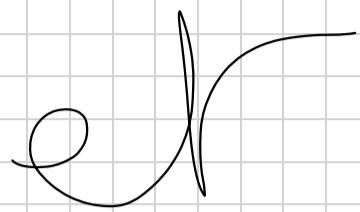
→ pour insérer easy rider, normalement se place c'est



mais il ya plus de place car max 4

et au min pour chaque nouvelle page 2

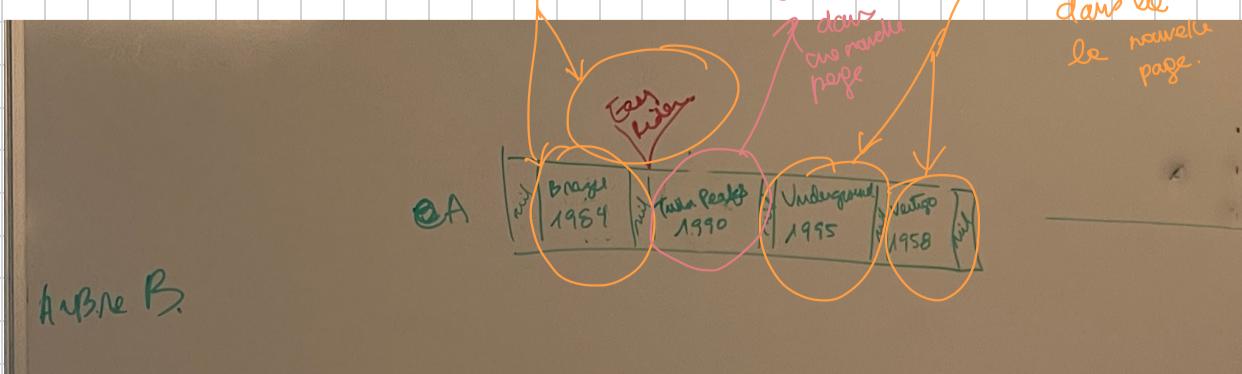
→ donc on alloue une nouvelle page:



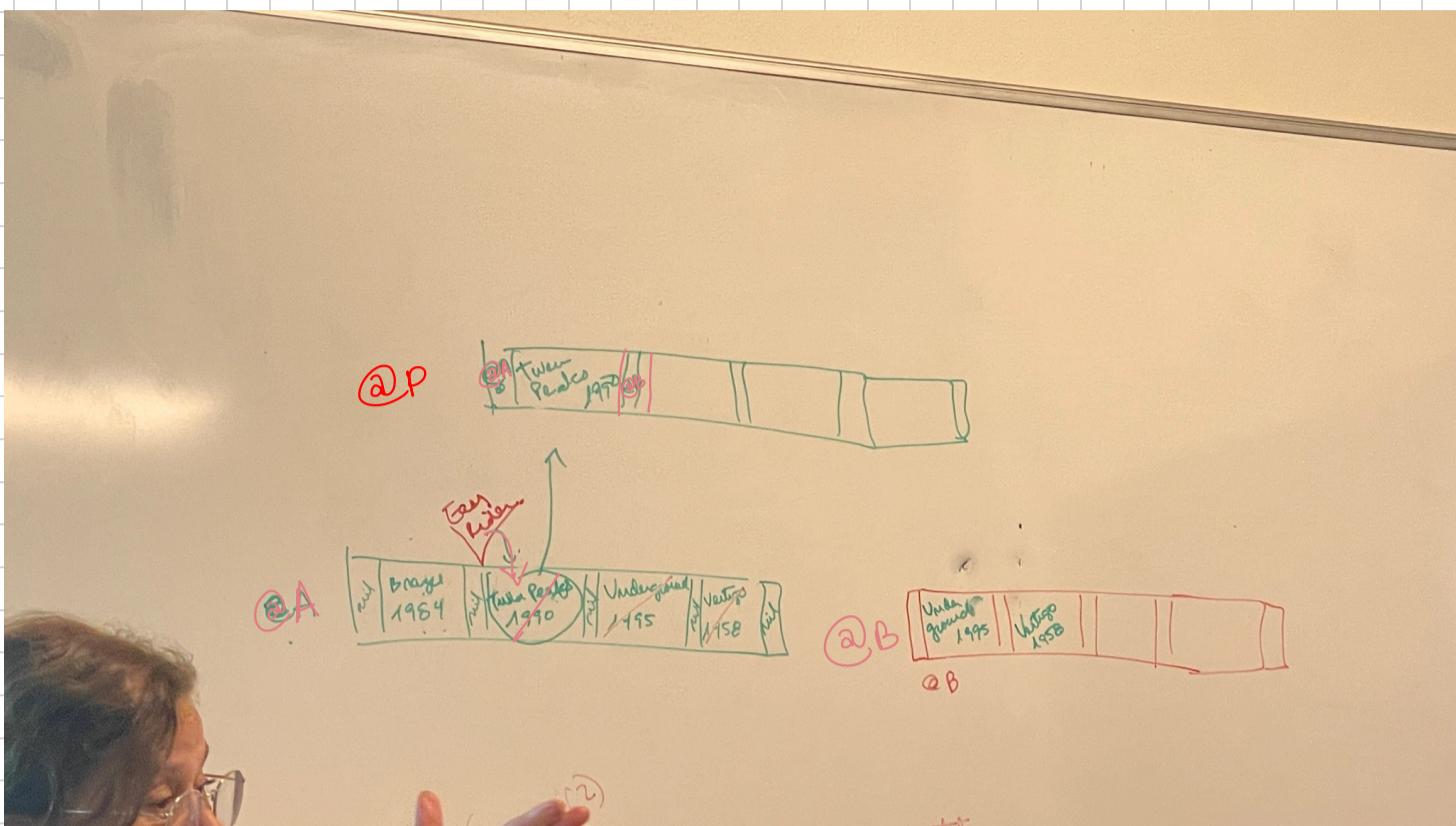
- on laisser dans la page A les m enregistrements avec les plus petites clés
- m'ici c'est l'ordre. = 2
- on déplace les m enregistrement avec des clés plus grandes à la nouvelle page et l'enregistrement du milieu il monte au niveau supérieur



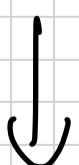
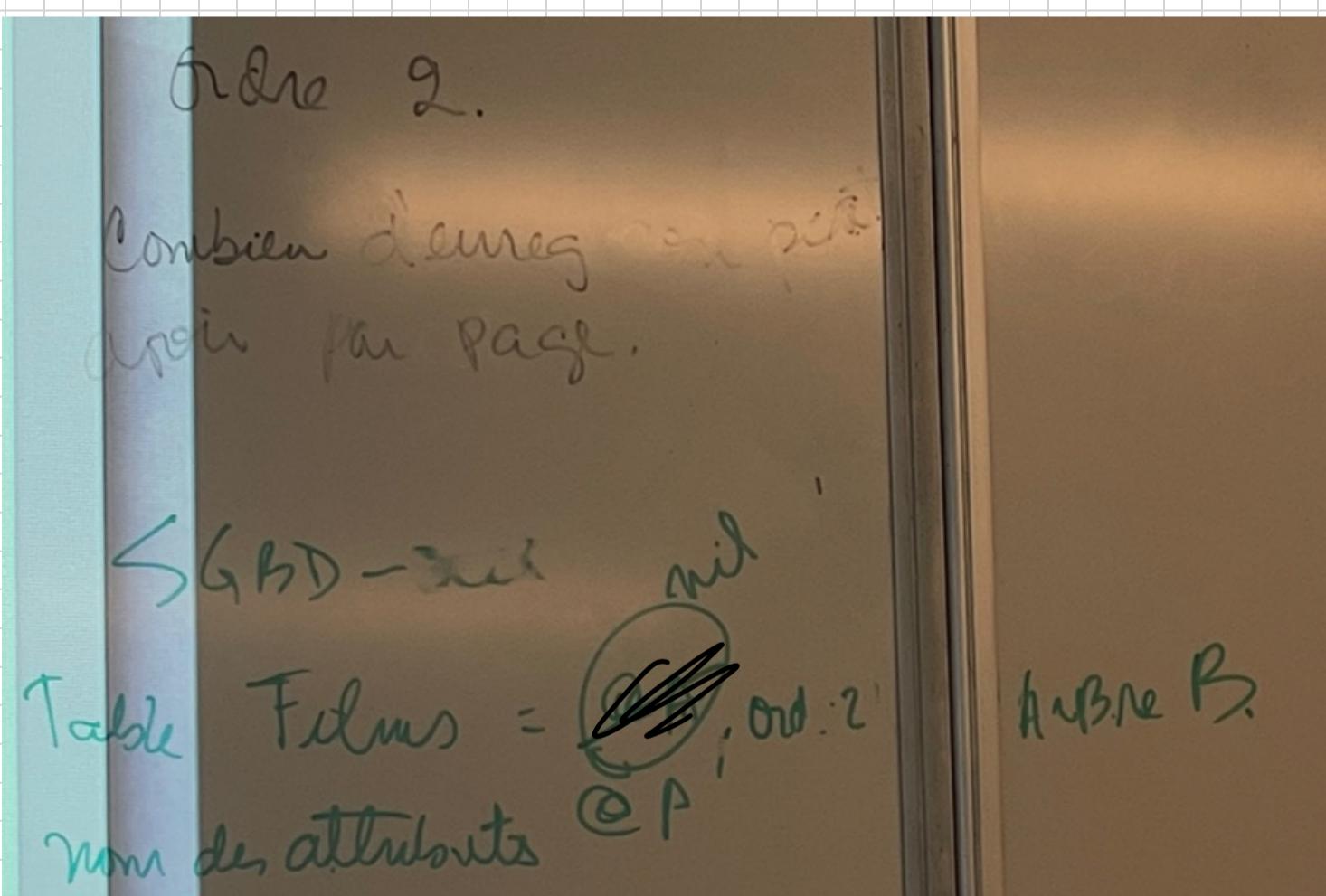
Ce donne Ce



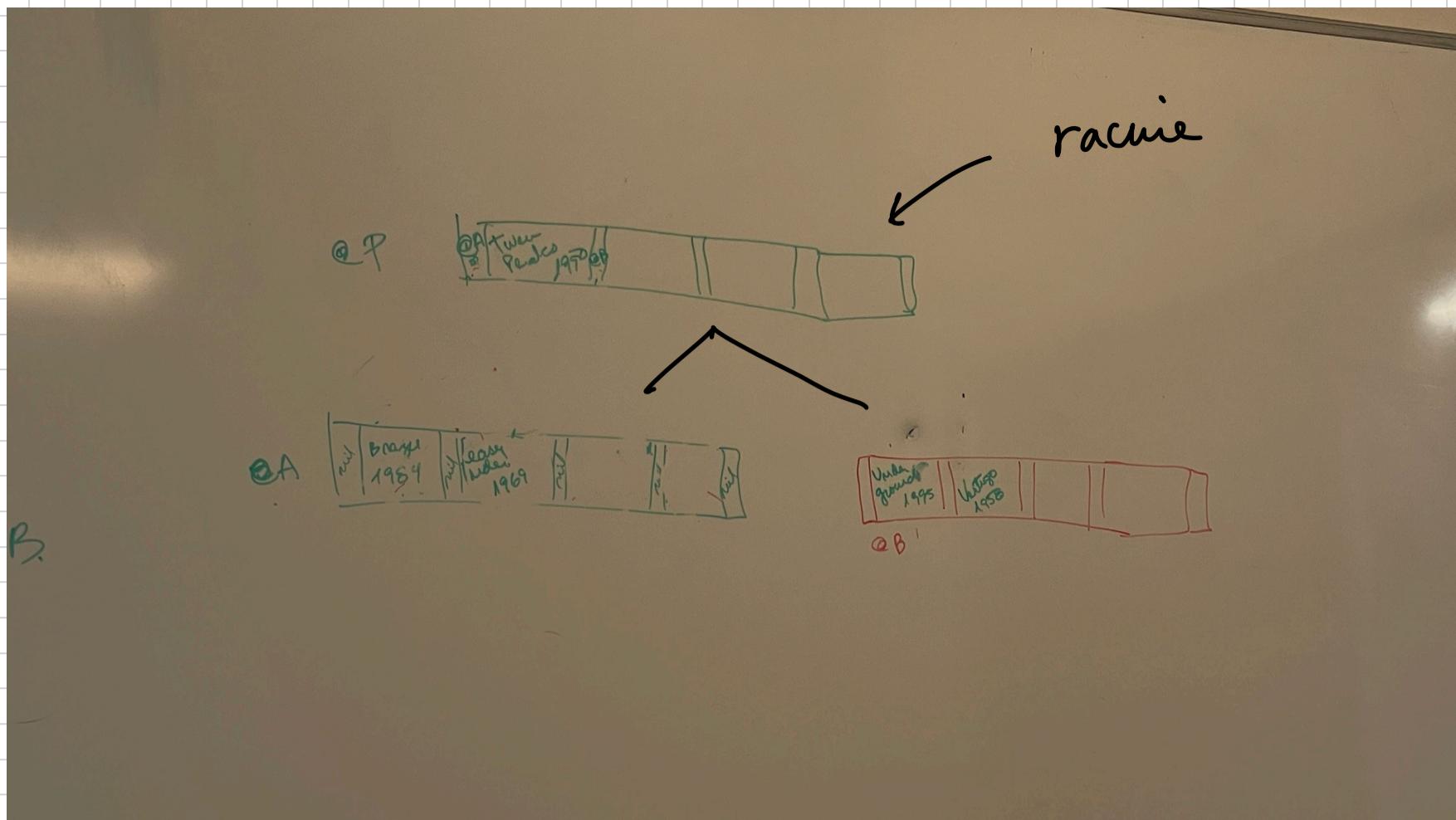
ce donne ce



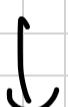
et dans les métadonnées : ce devient comme ce



en plus clair



Pr inséner psychose



on va dans les métadonnées

on trouve l'adresse  
dans le grecin

c'est @ P



On va dans le page qui  
a l'adresse @ P

on comprend le P de psychose  
avec le T de Twin Peaks



ordre alphabétique



↓

P avant T

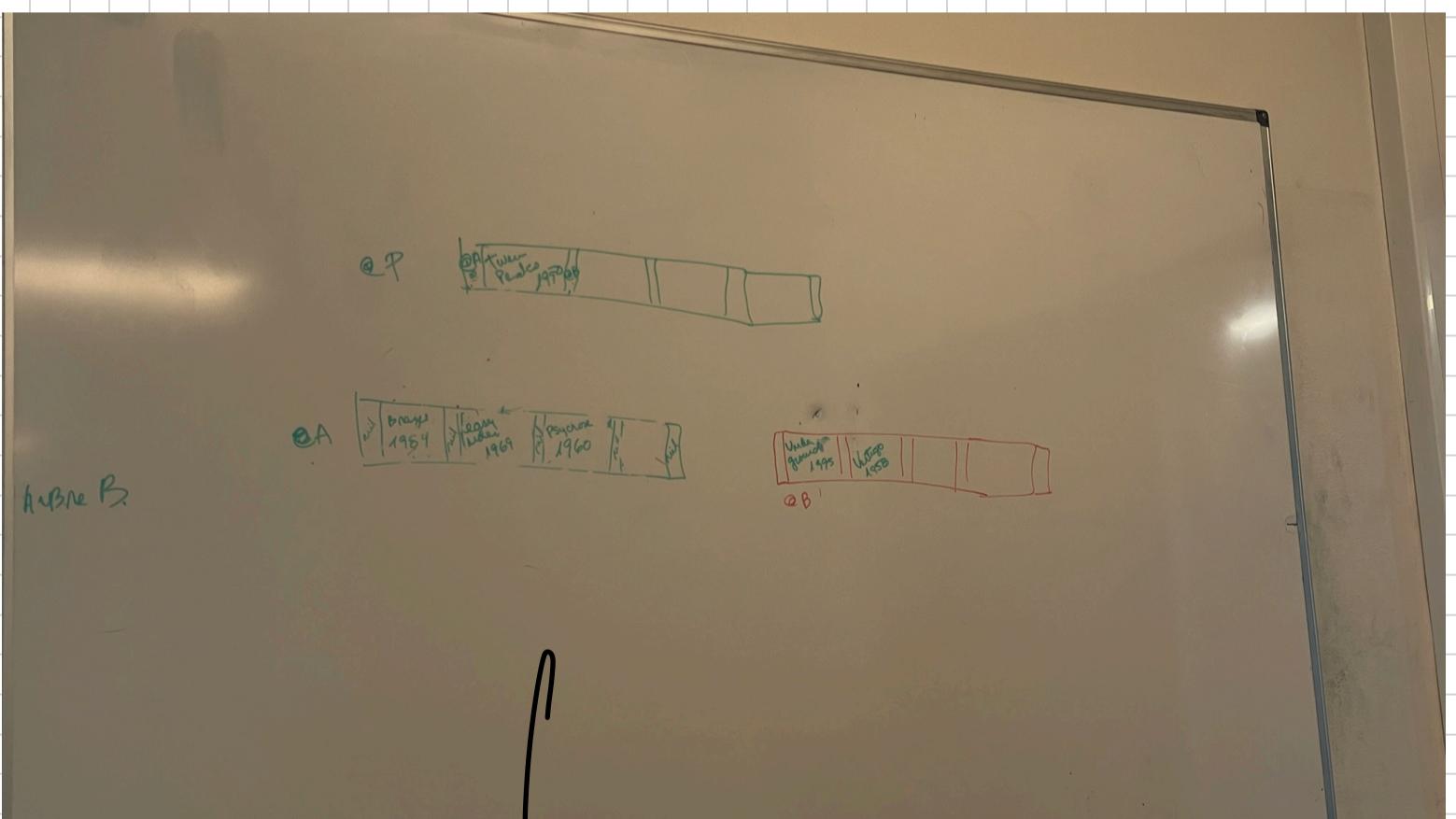
↓

donc on va à

@A

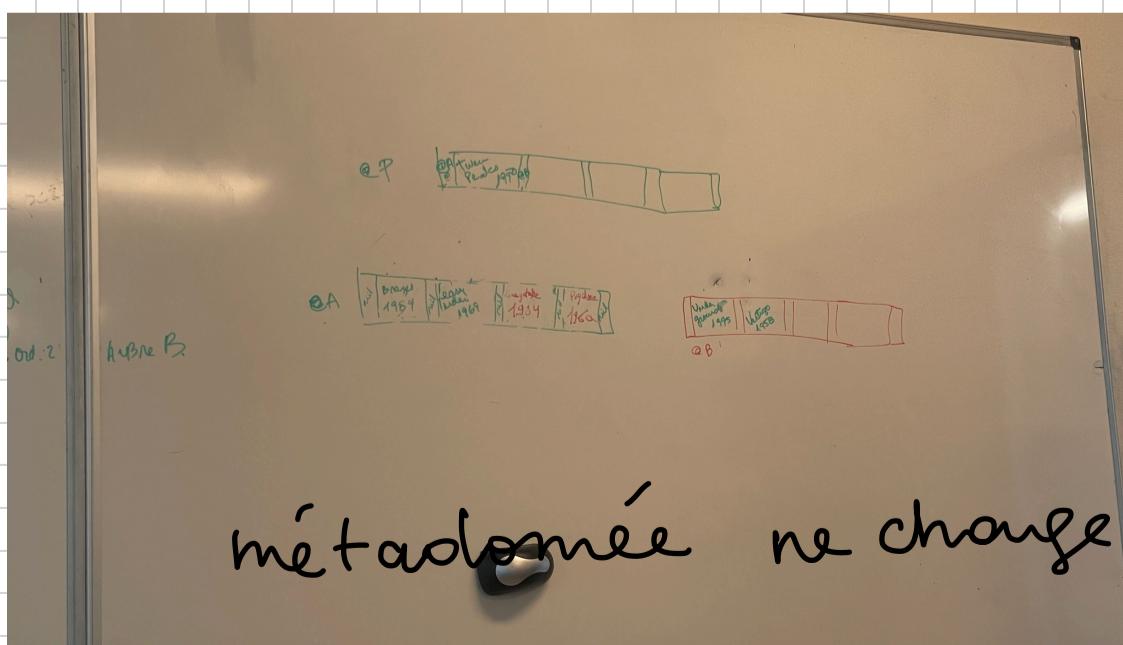
↓

et on l'insère au  
bon endroit



on insère grey stock

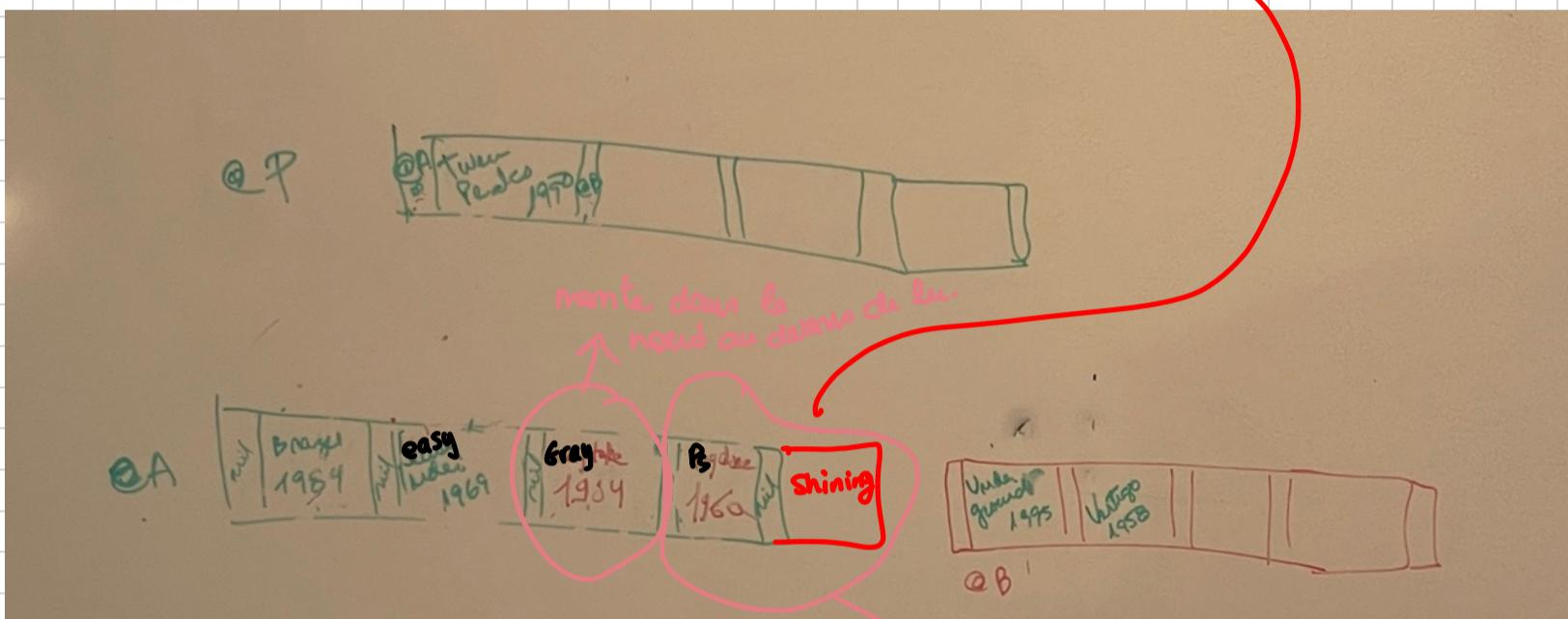
(je skip les étapes)



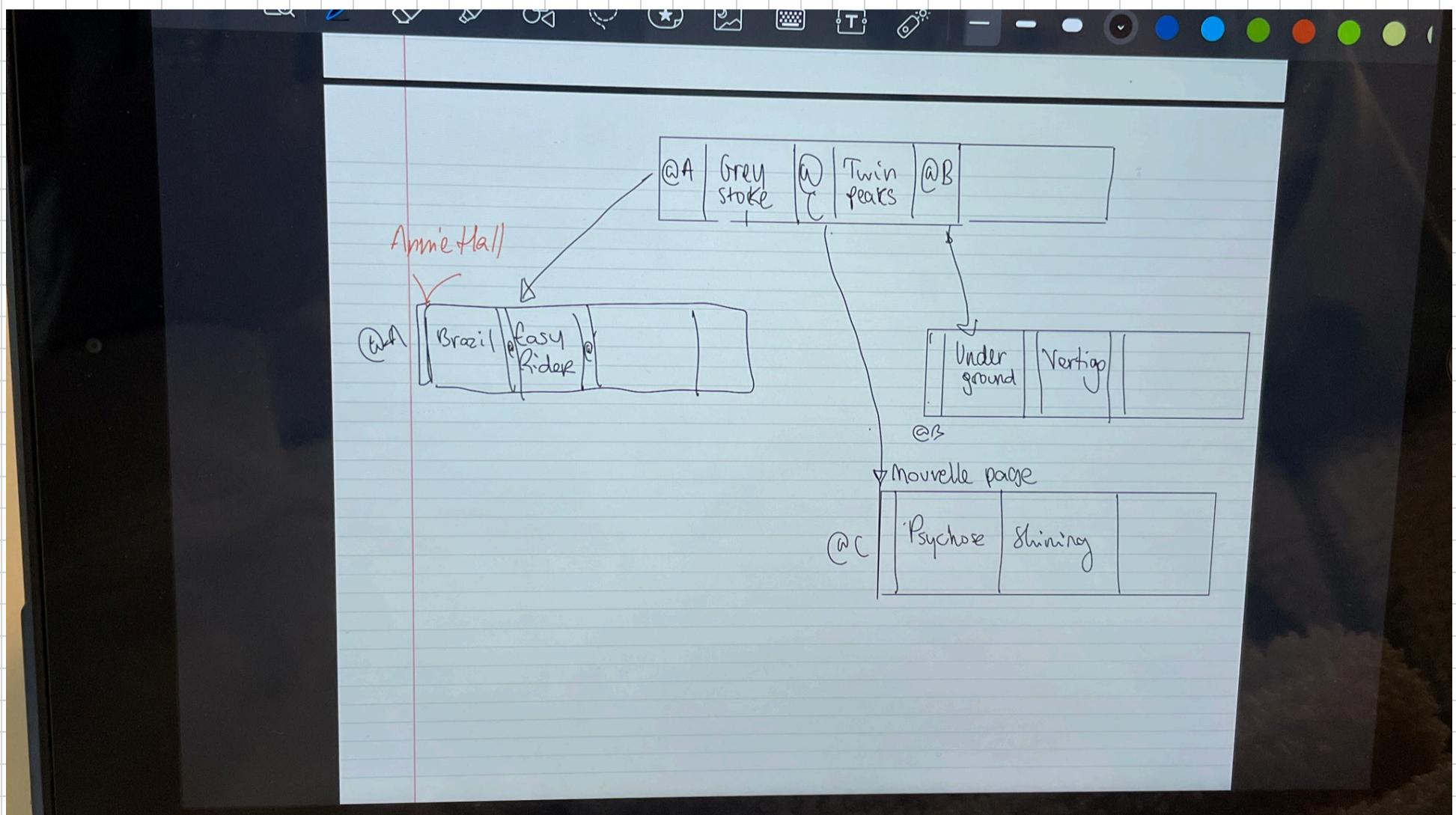
métasomée ne change pas

on insère Shining (je skip les étapes)

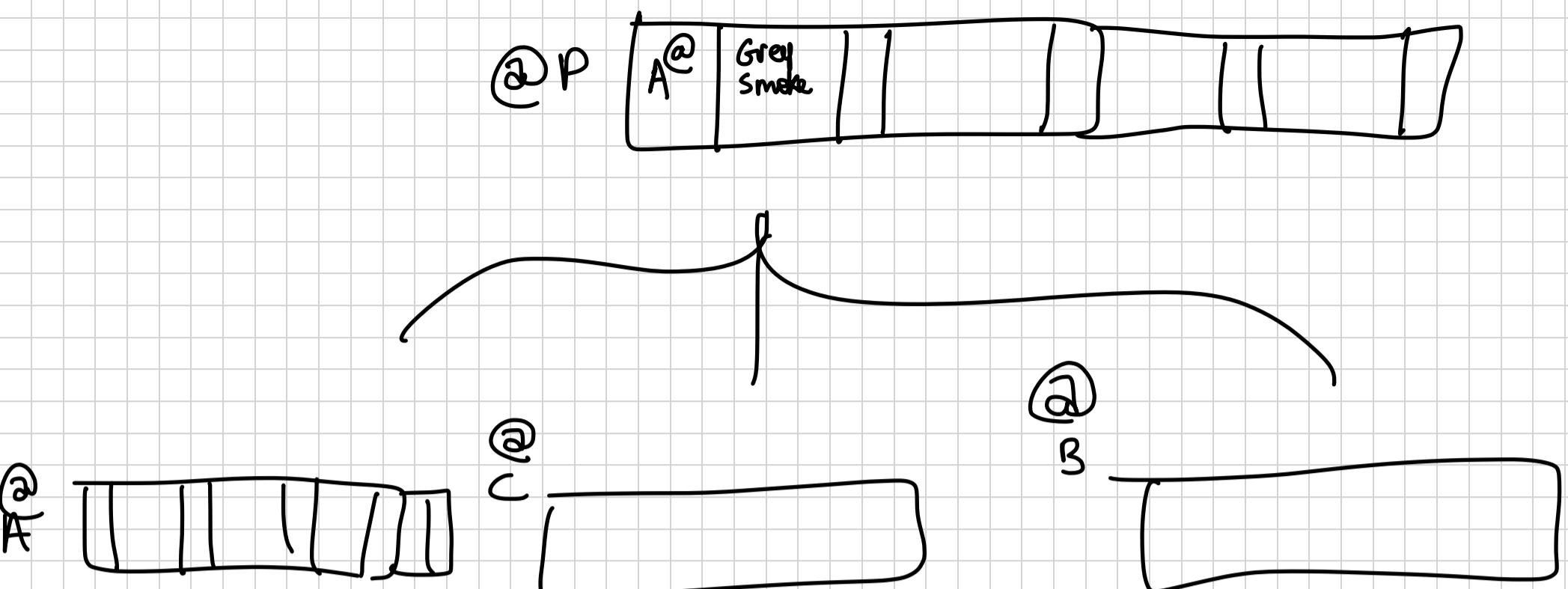
normalement il va à l'ordre alphabétique.



on les déplace  
dans une nouvelle  
page @C



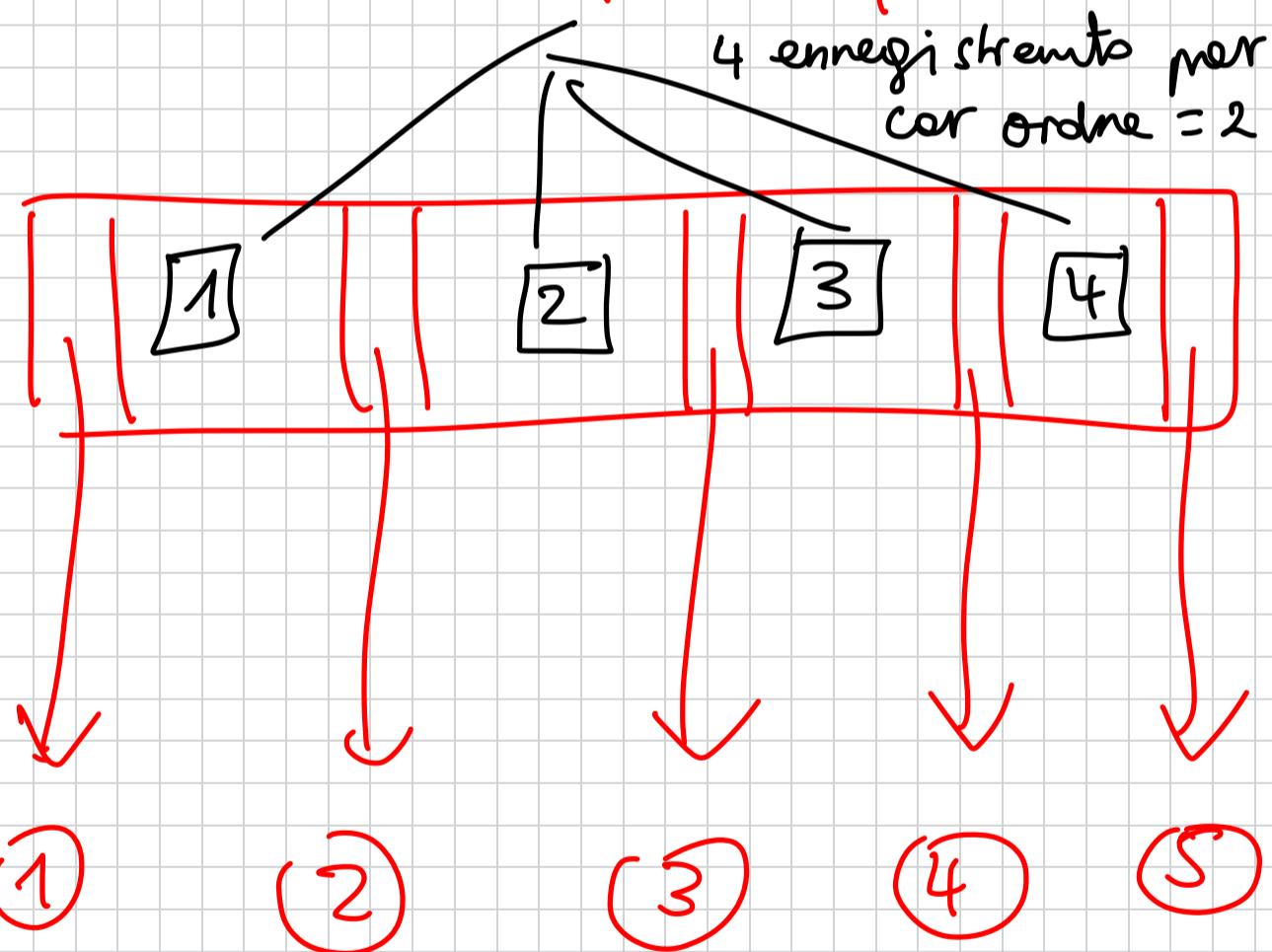
explications



@A @C @B sont au même niv.

Sur un même niv on peut peut avoir  
5 pages

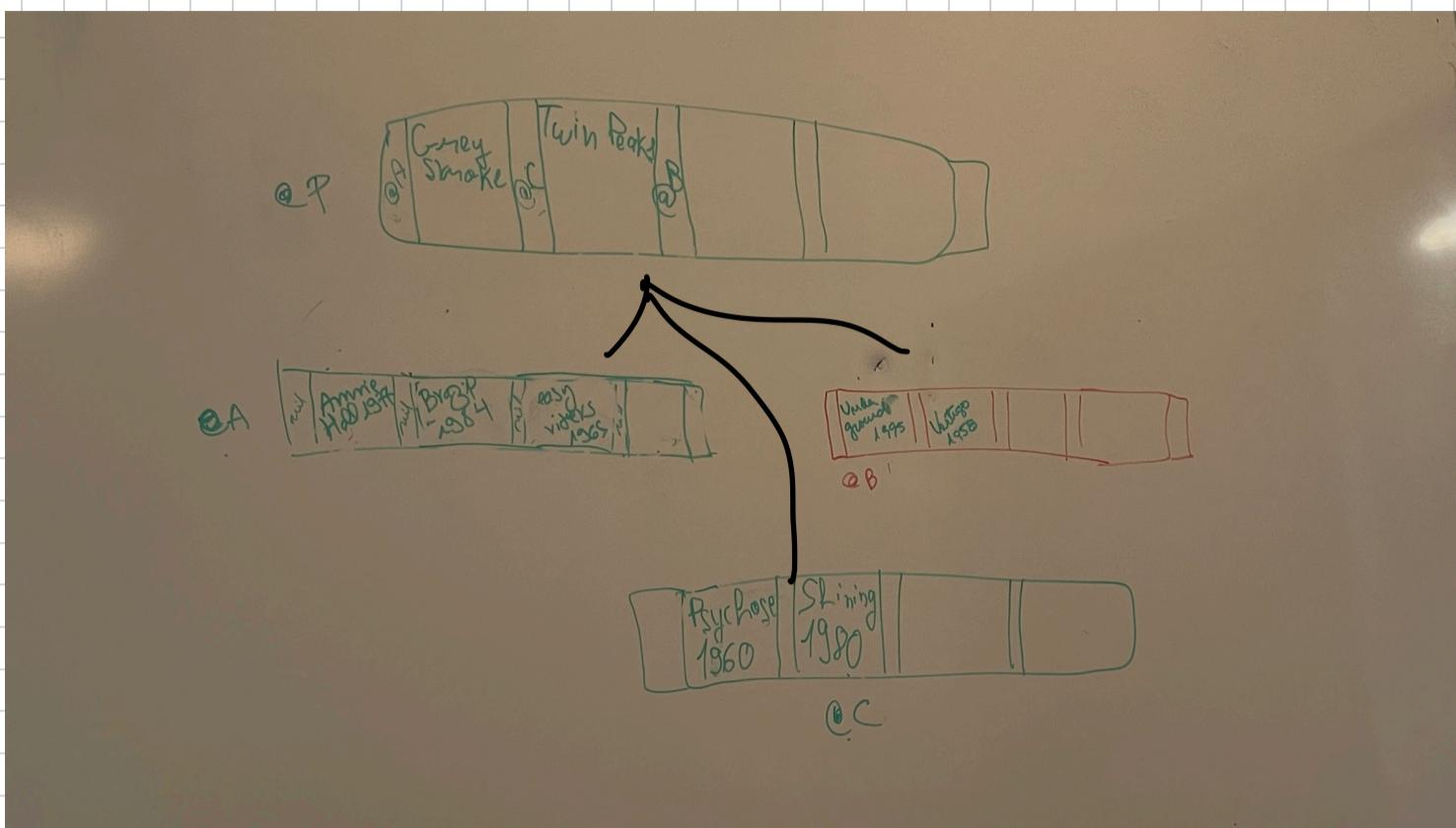
4 enregistrements par page  
car ordre = 2



5 fils au max

Arbre B ≠ arbre binaire

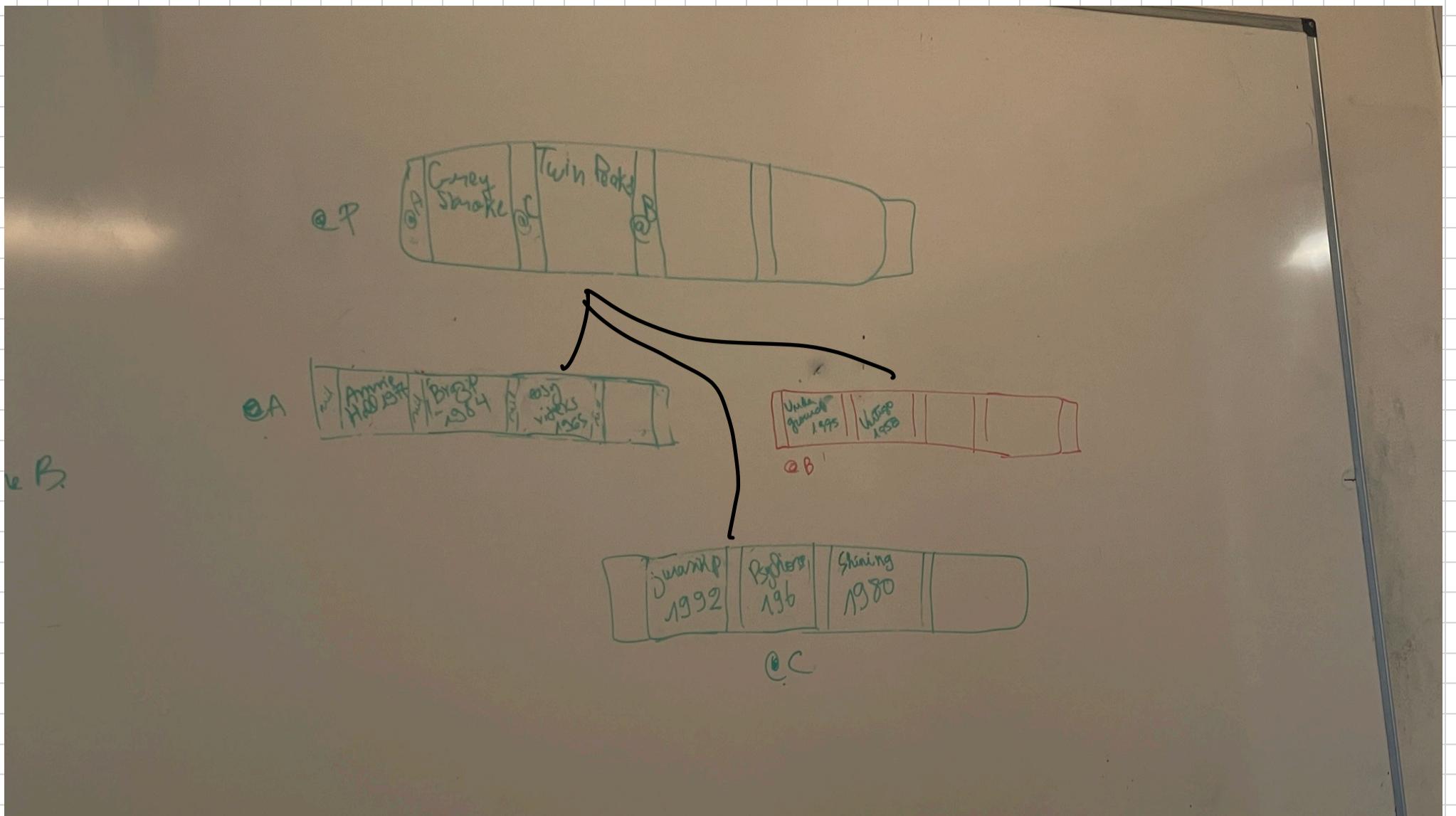
↓  
on insère Annie Hall



↓  
on insère  
Jurassic Park

Rappel : quand on insère, on insère dans les feuilles (le niv le plus bas dans l'arbre)  
↳ c'est les feuilles qu'on doit replier et si c'est replié on crée de nouvelles feuilles

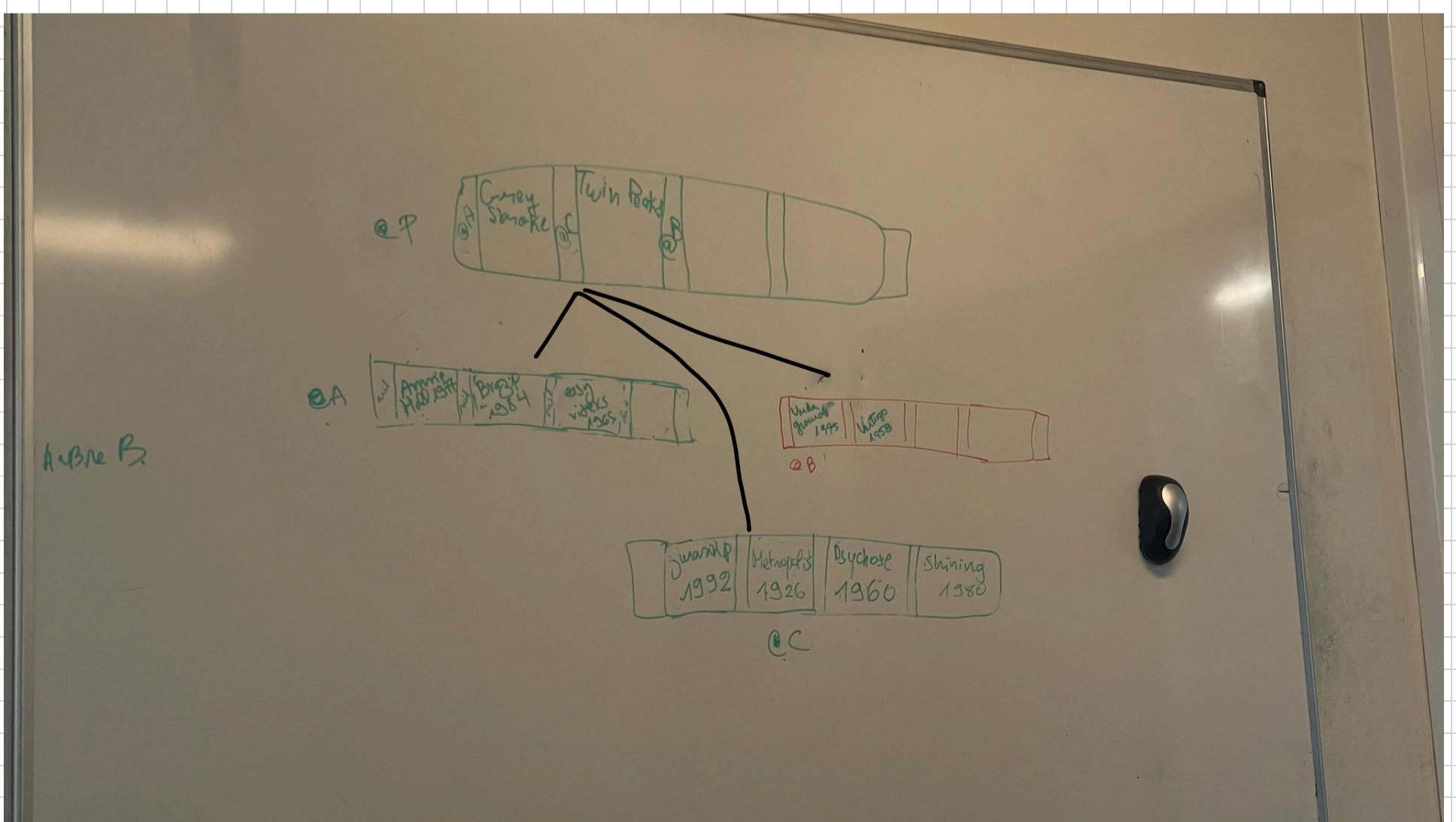
↓  
ce donne ce



Rq: quand on insère,  
on enregistre un  
enregistrement en  
entier = titre + année

↓  
On insère metropolis 1926

il va dans @ C entre. psychologie et  
jurisprud.



## À ce niv

- Si on veut chercher assistance, on fait 2 lectures  
↳ il y existe pas encone
- ↳ " n'importe quel enregistrement  
on doit lire → max : 2 (la racine + une feuille  
(c'est que deux niveaux))  
→ min : 1 page (la racine)  
(si ce qui on cherche est  
dans la racine)

Rq: quand le prof dit la clé de la sélection  
elle parle de la clé primaire de la table

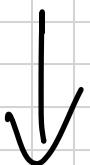


On insère mon hutan

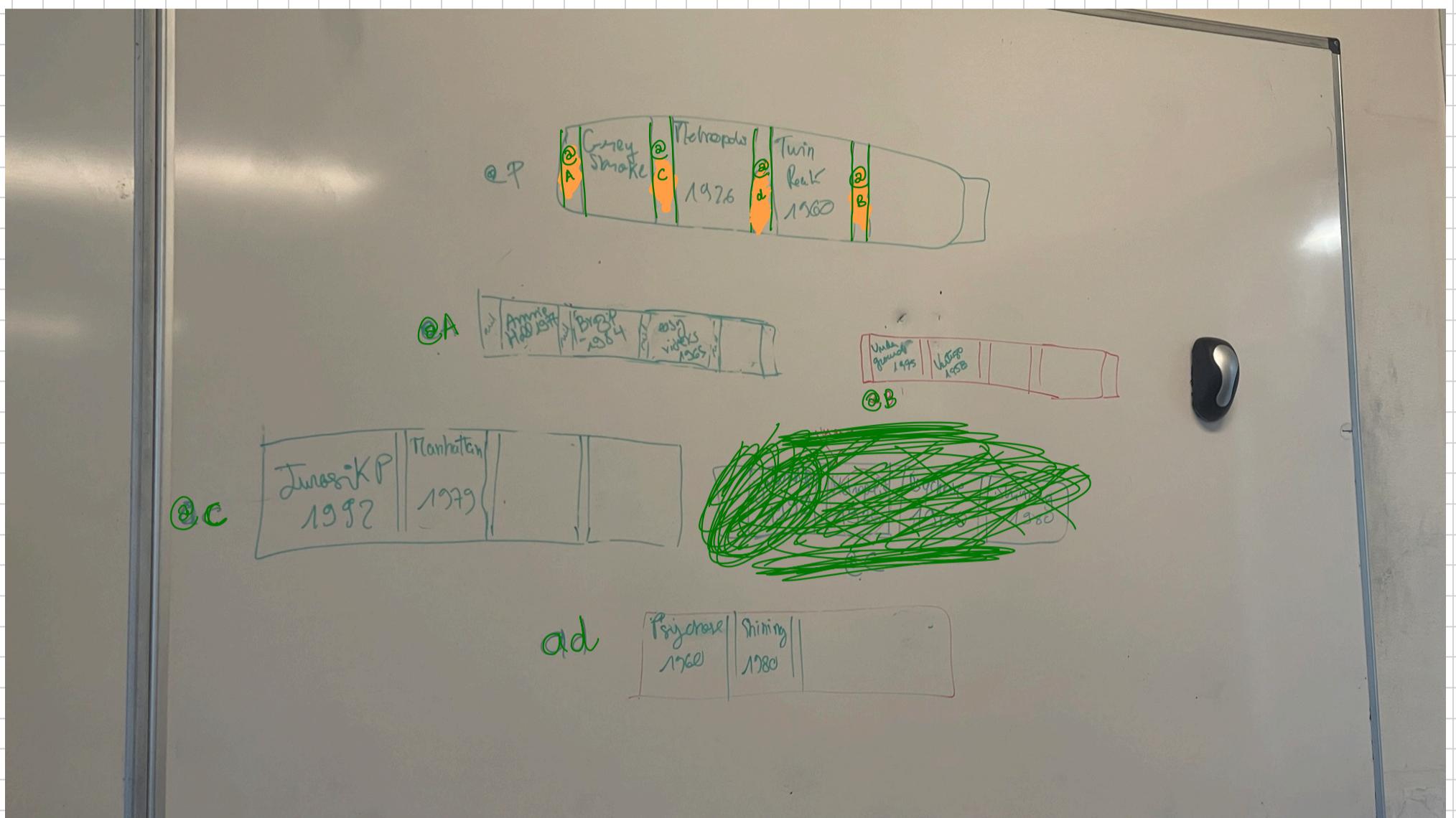
• on monte le milieu

• on met les deux derniers dans une  
nouvelle page

• on laisse les deux premiers dans la 1<sup>re</sup> page



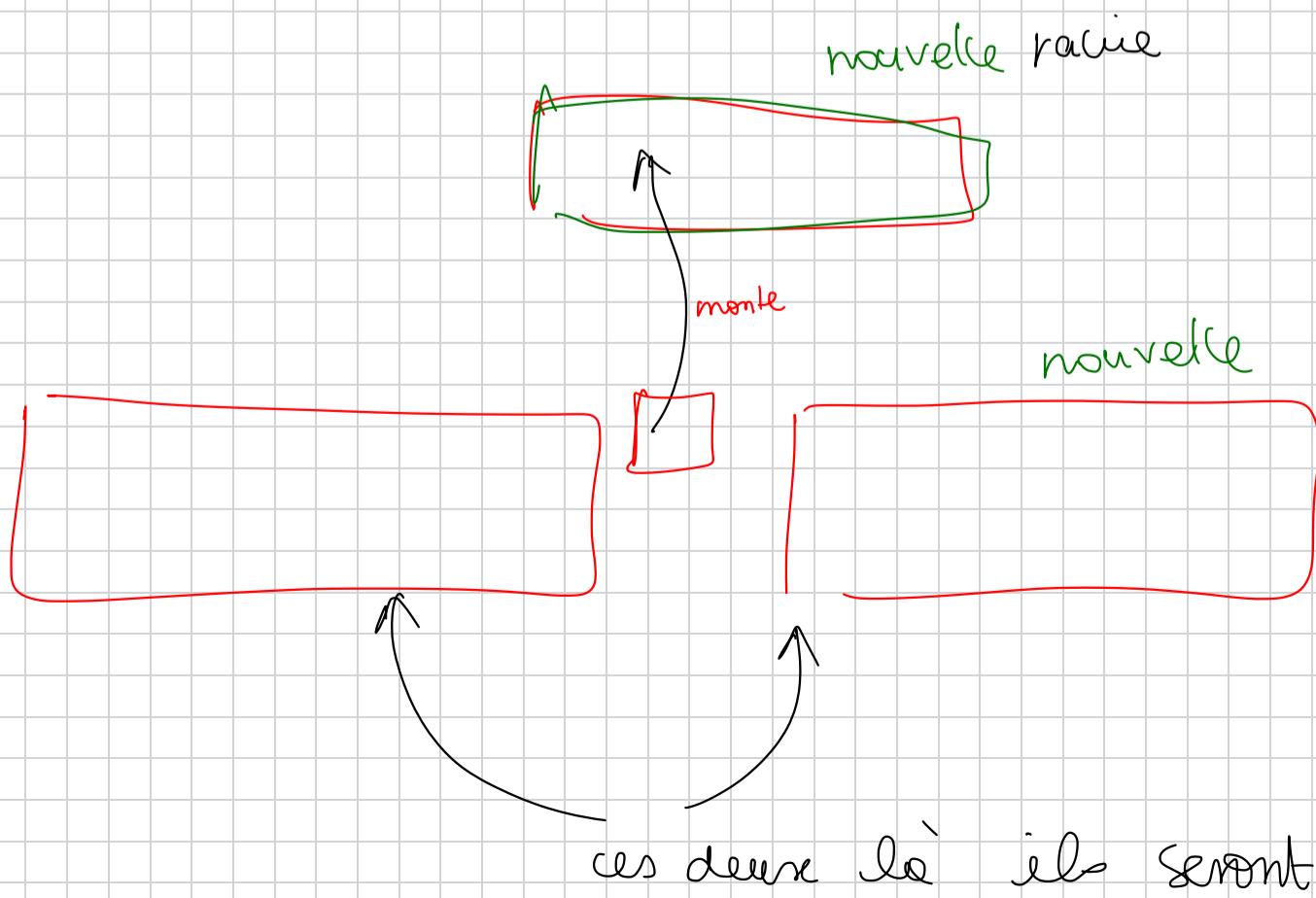
ça donne ça :



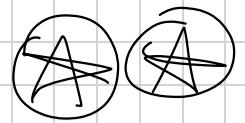
le niv de l'arbre = on compte le racine dedans

Rq: si le racine est groupé, on crée 2 Pages (correspond à pas de dessus le racine)

↳ on éclate la racine



équilibrés → c'est à dire

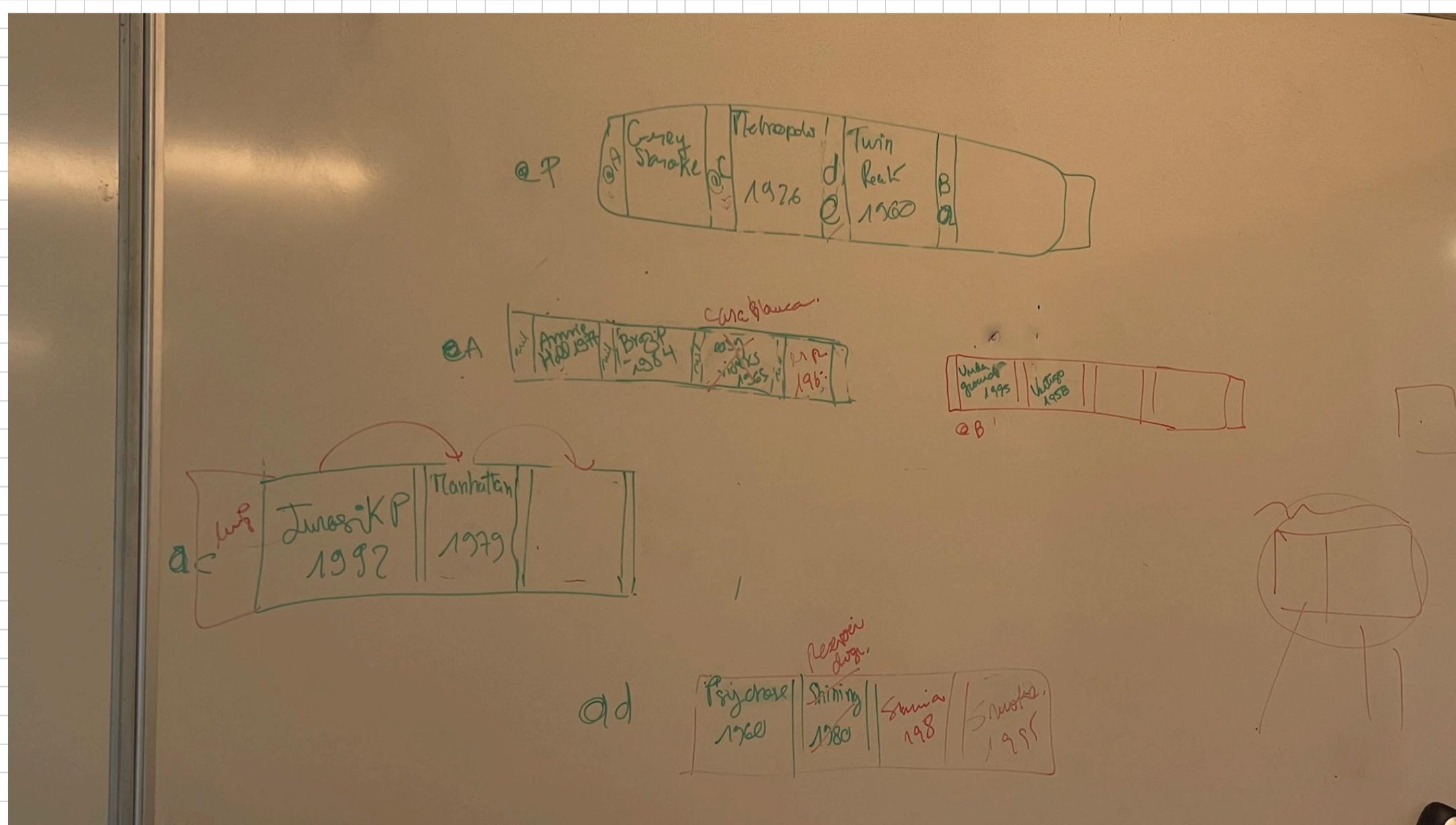


que le diff entre  
les deux c'est max 1



= c'est l'arbre équilibré

↓  
on insère  
le reste



la structure du fichier = comment on a shocké → en arbre

B

Réponse à la question du TD : nbre de pages : 5

il ya un ordre dans ce qu'on a fait

@A → greysome puis @C puis metropolis puis le reste

1. B) on le fait rapidement

de l'arbre

• Ordre  $\downarrow 2 \rightarrow 4$  clés  $\rightarrow 5$  adresses

• on suppose que dans une page on peut stocker

4 enregistrements  $\leftarrow$  attention il faut que ce soit pair  
on peut pas mettre un 3 par

ⒶⒶⒶ

~~pair~~

ⒶⒶⒶ

→ si on nous donne la taille de l'enregistrement et  
la taille d'une page on aurait pu le calculer  $\rightarrow$  si c'est en  
mais là on nous a rien donné donc on le suppose 4

• on le fait sur l'attribut titre (presannée comme dans l'énoncé)

• on va stocker suivant un arbre  $B^+$

↳ on stock tjs du bas vers le haut dans  $B^+$

le 1<sup>er</sup>:

on insère vertigo

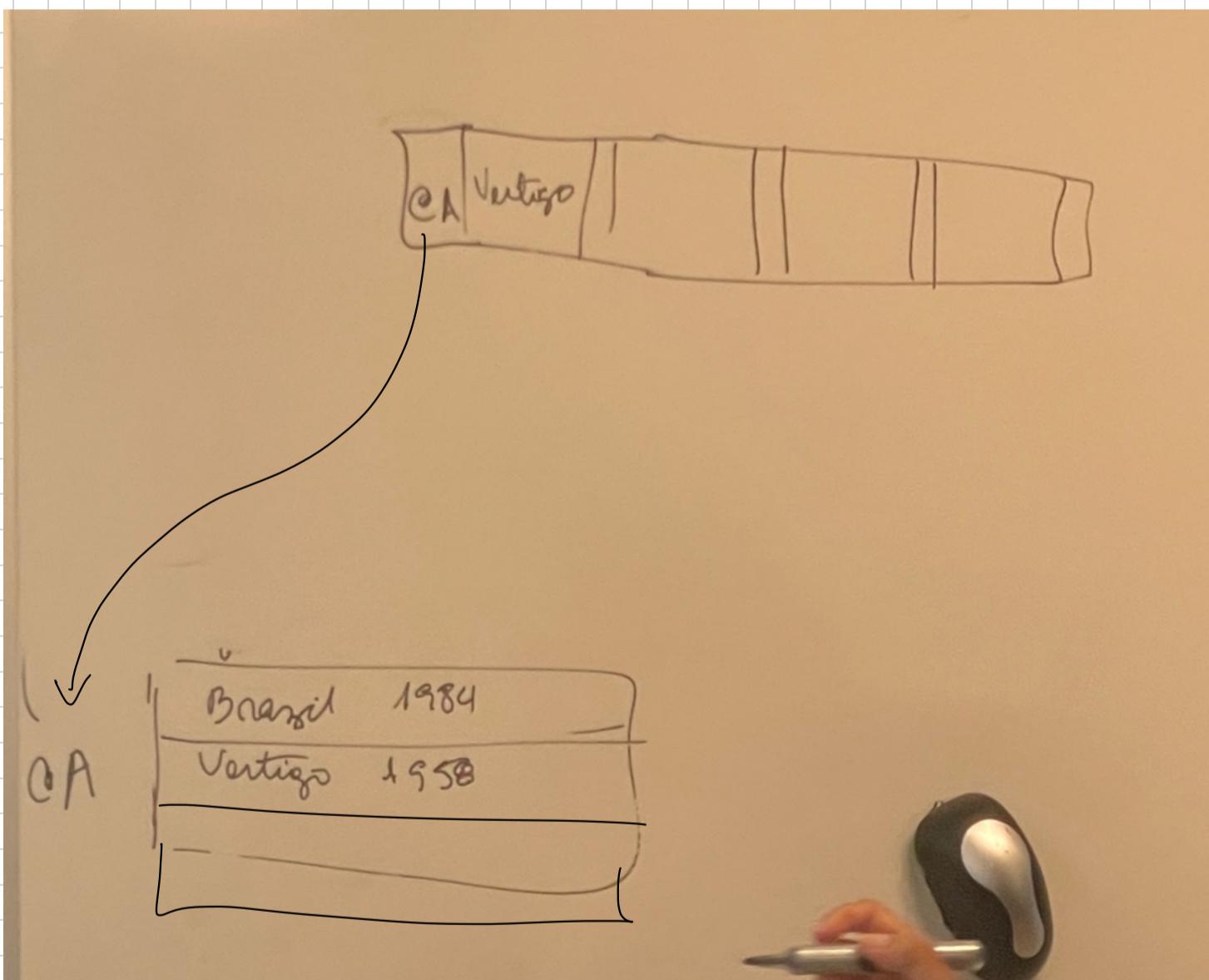
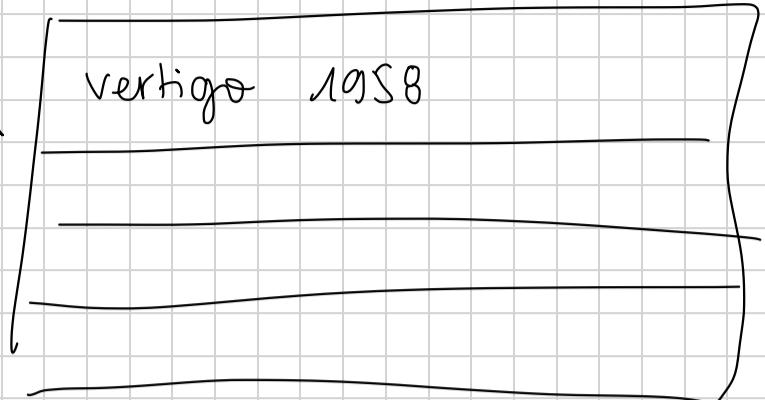
comme clé dans le racine

comme enregistrement

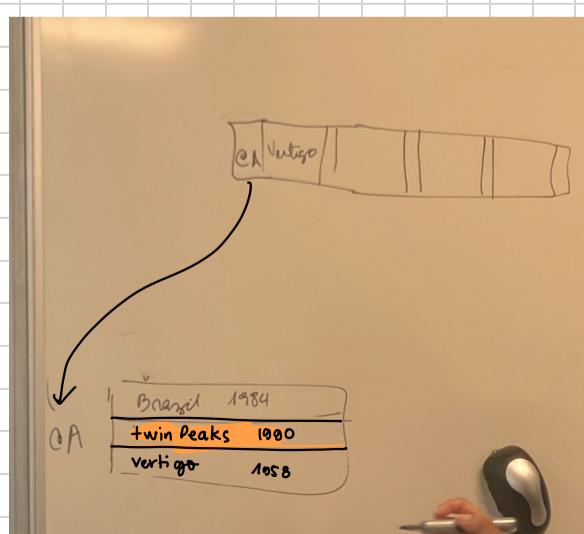


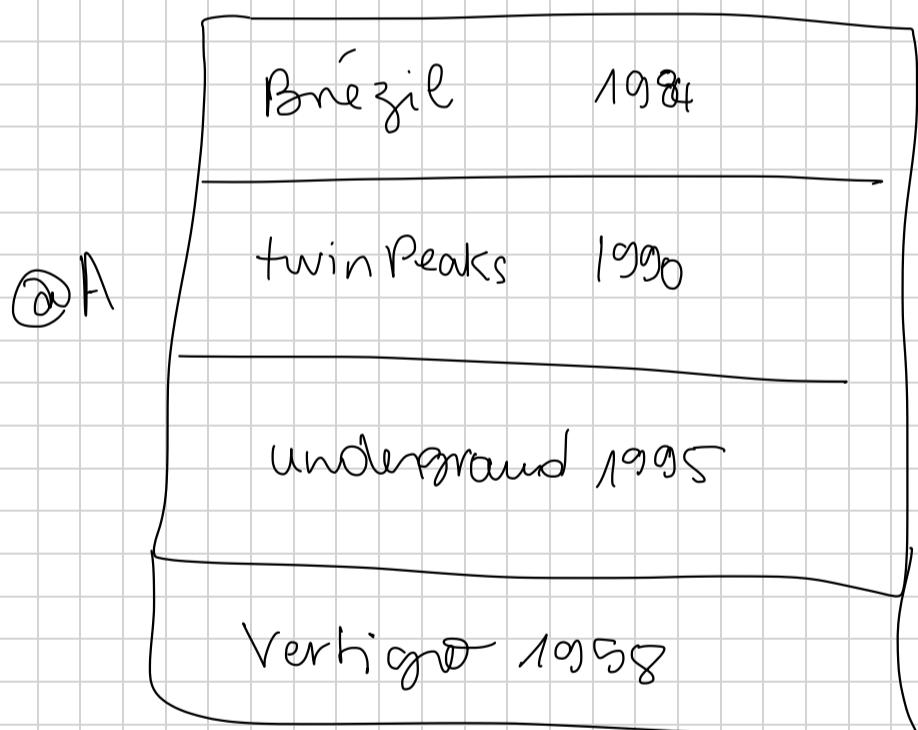
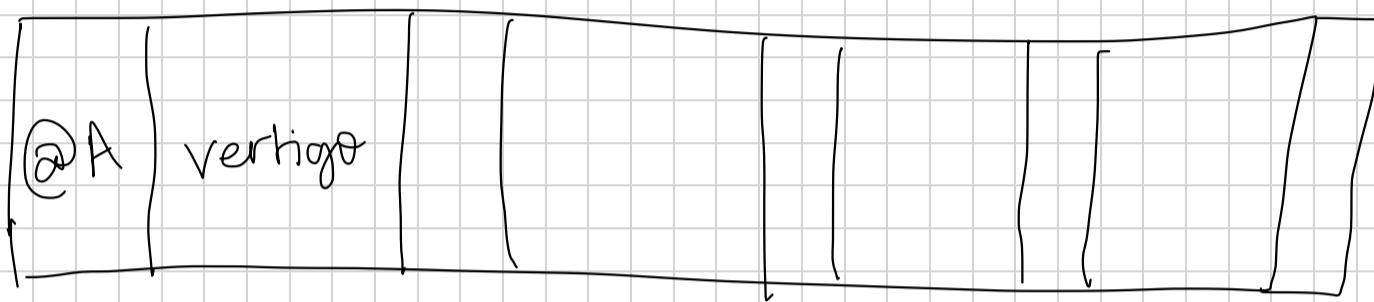
tous ce qui est  $\leq$  vertigo

@A

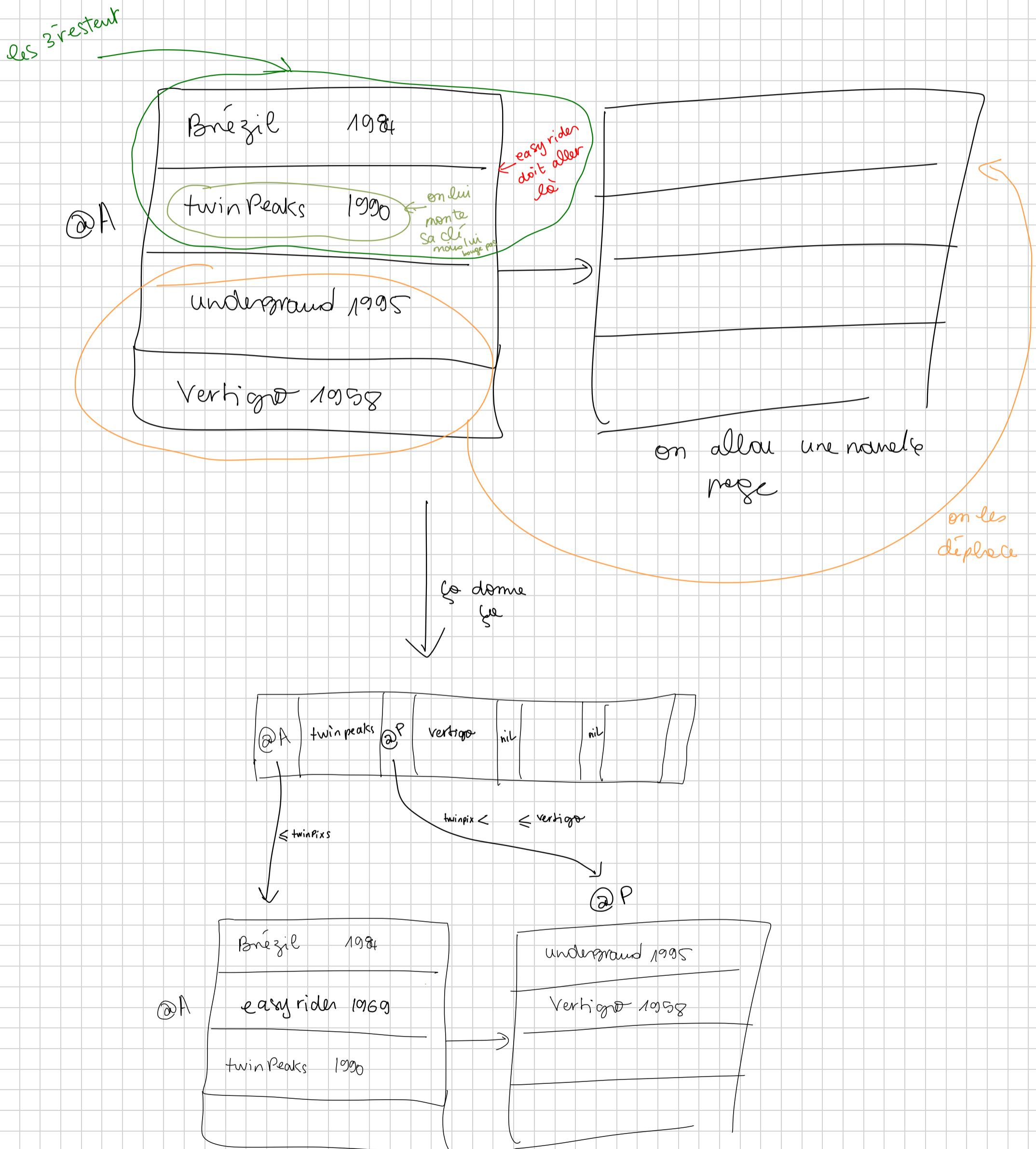
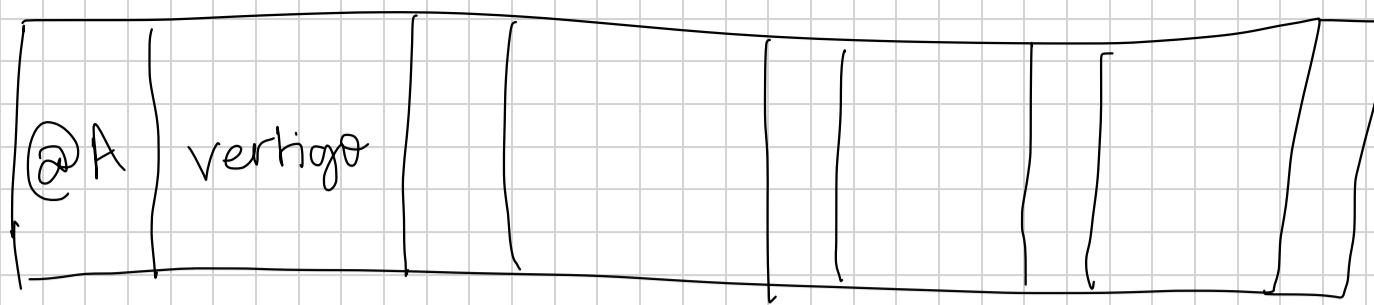


↓ on inside twin P1X





on insère <sup>easy rider</sup>, il est plus petit que vertigo  
 → plus de place → on alloue une nouvelle page → et laine les 3 premiers  
 on décale les 2 derniers et le milieu  
 on monte sur clé



on insère psychose : ya de la place .



≤ twinpixs

twinpix < ≤ vertigo

@P

@A

Bnégisil 1984

easy rider 1969

psychose 1960

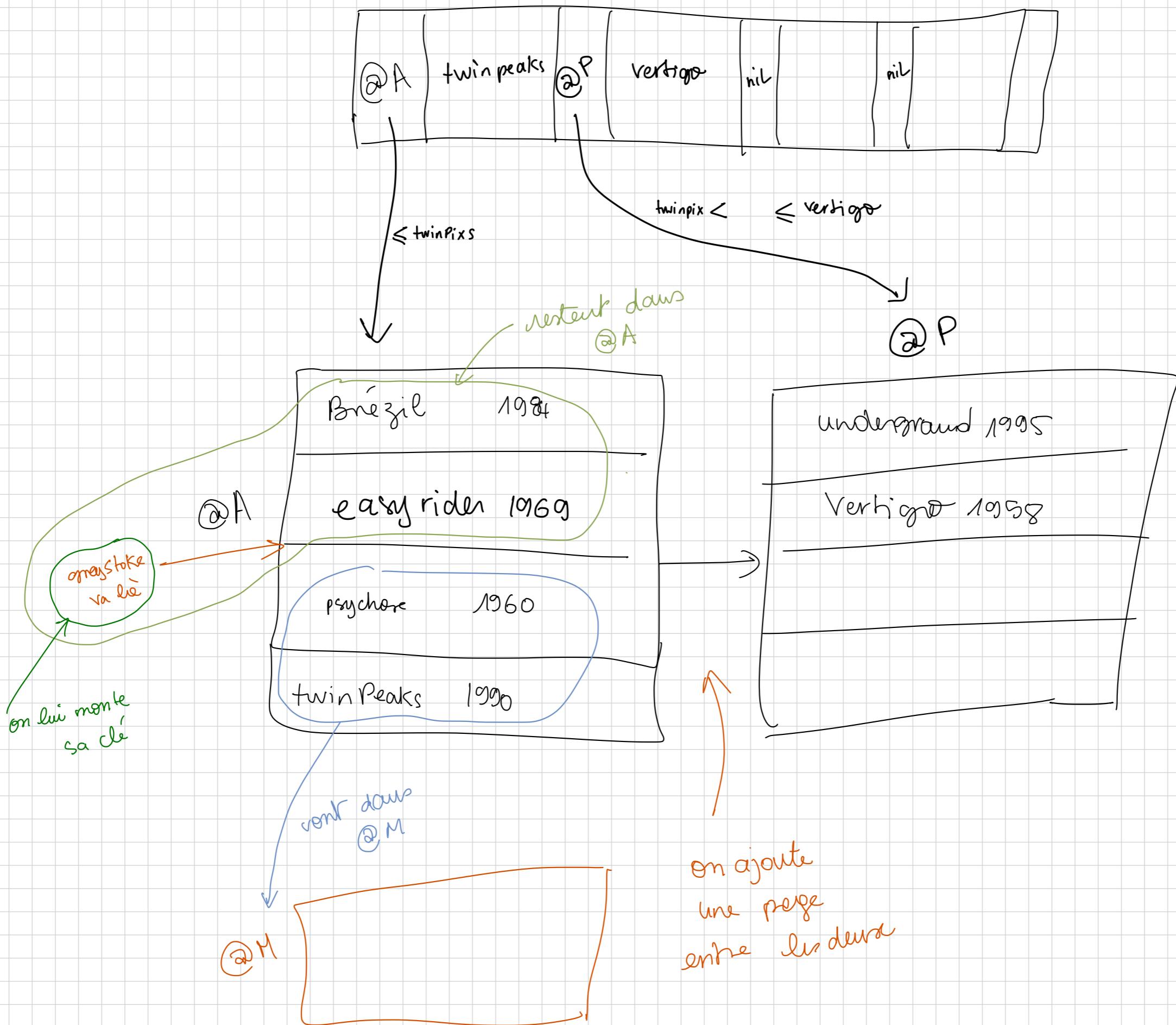
twin Peaks 1990

underground 1995

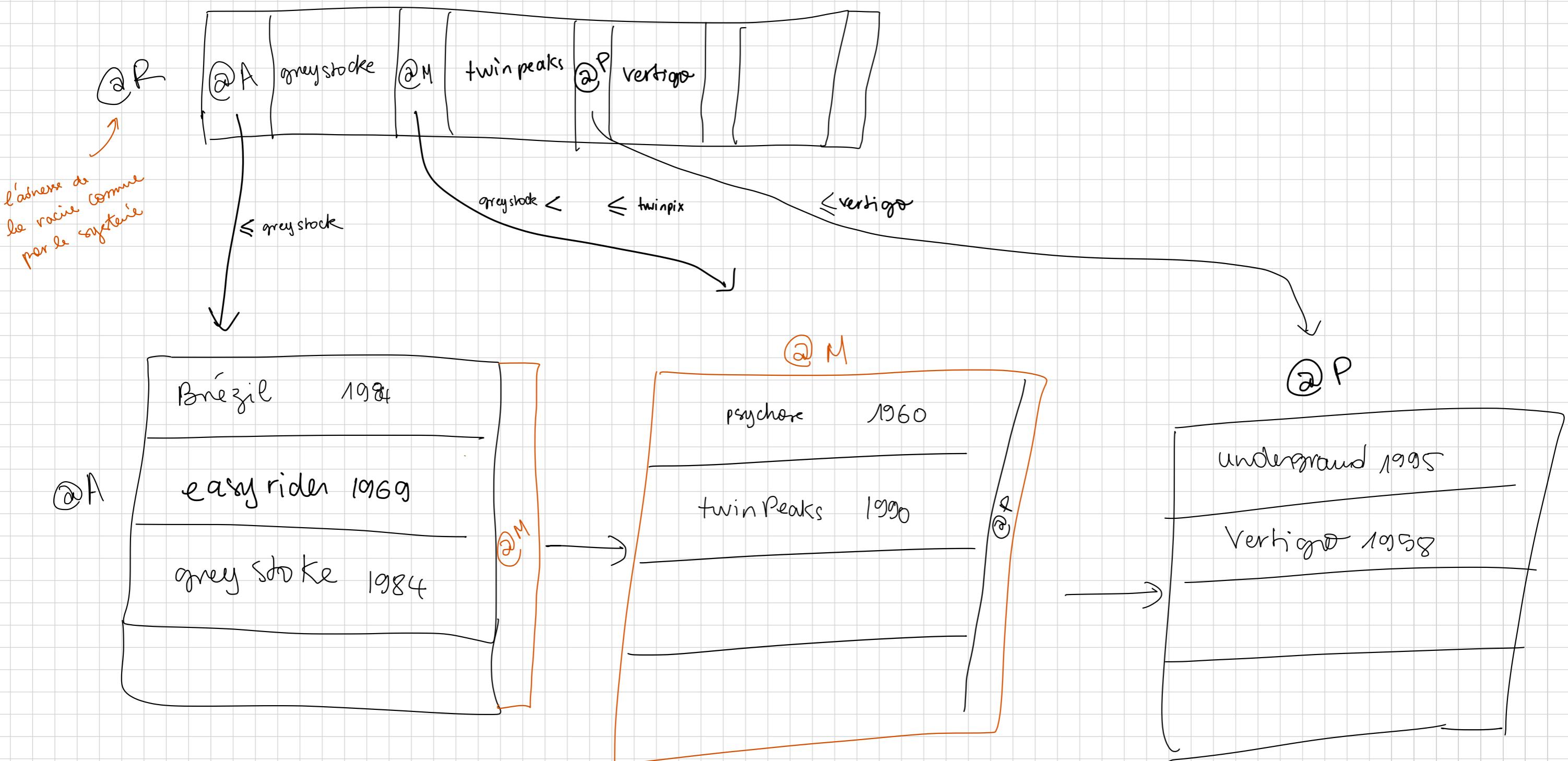
Vertigo 1958

on insère grey storme

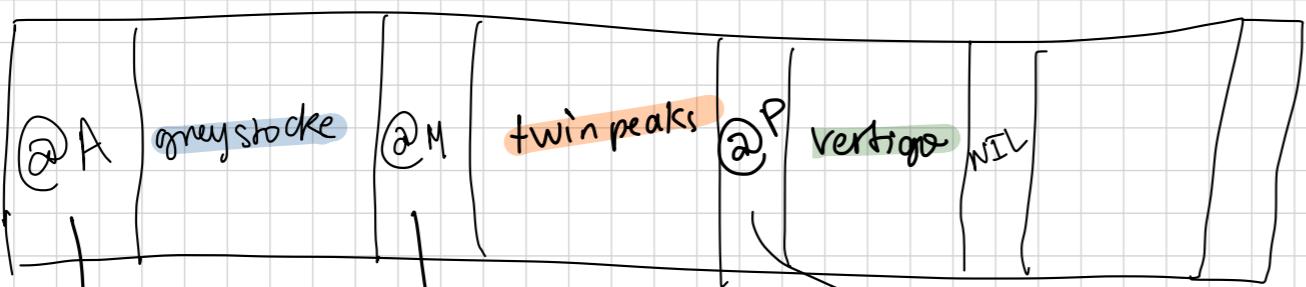




le dome le



on ajoute shining



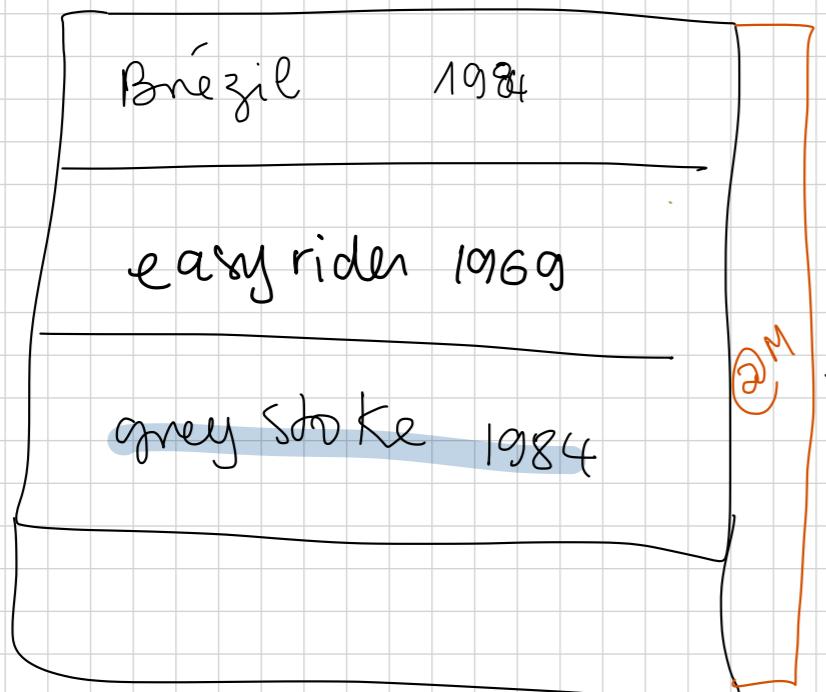
≤ grey stock

grey stock <

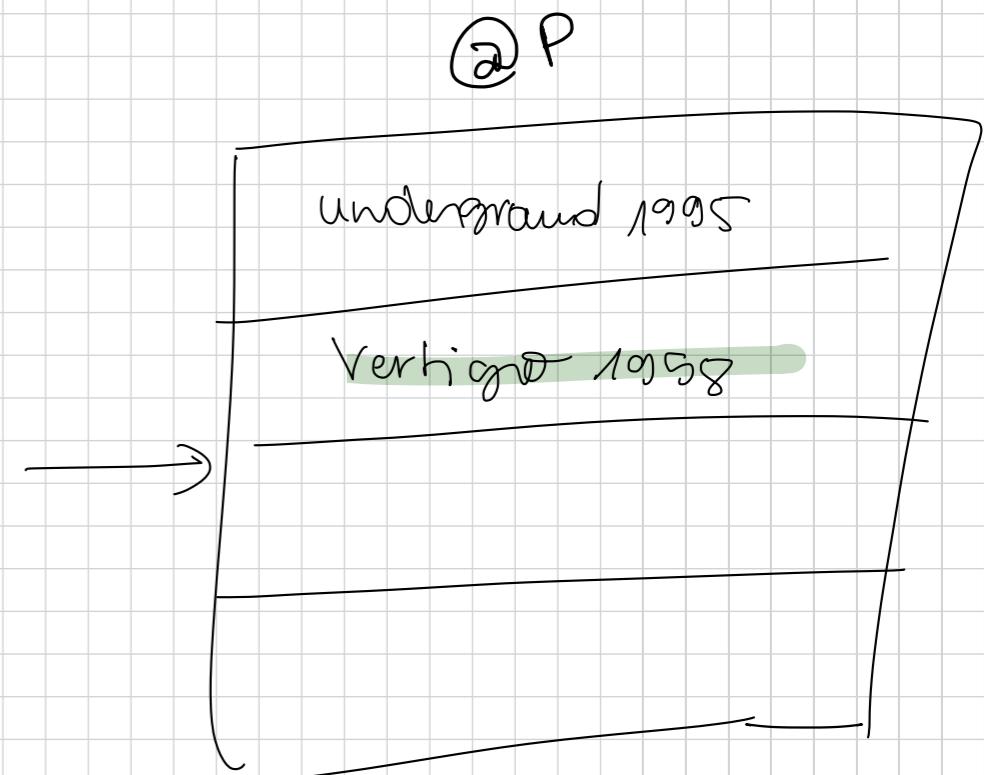
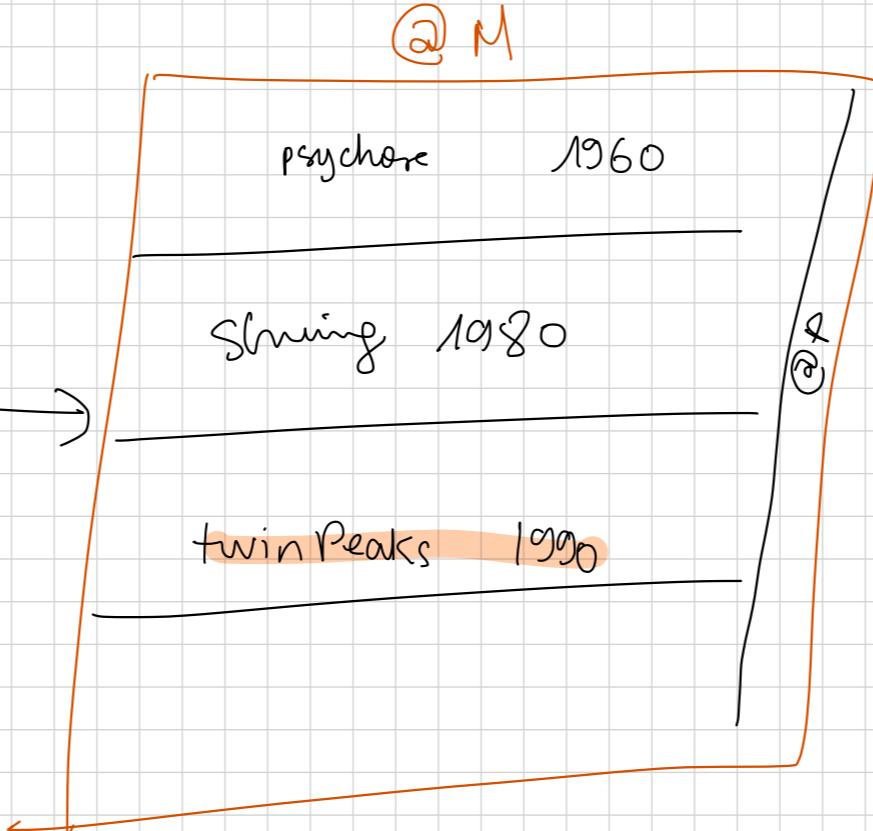
≤ twinpix

≤ vertigo

@A

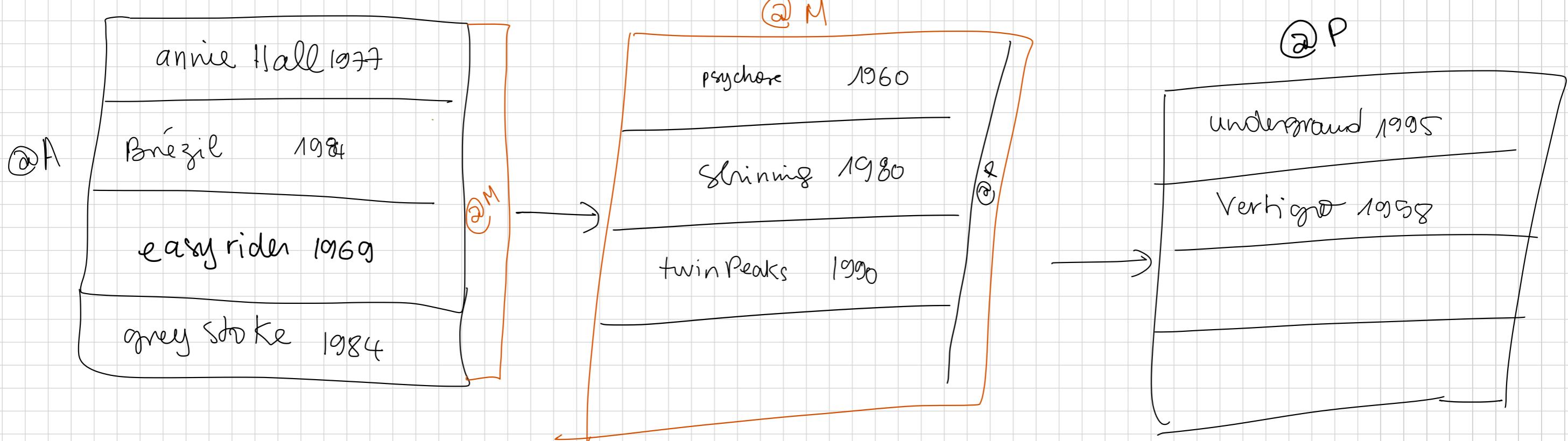
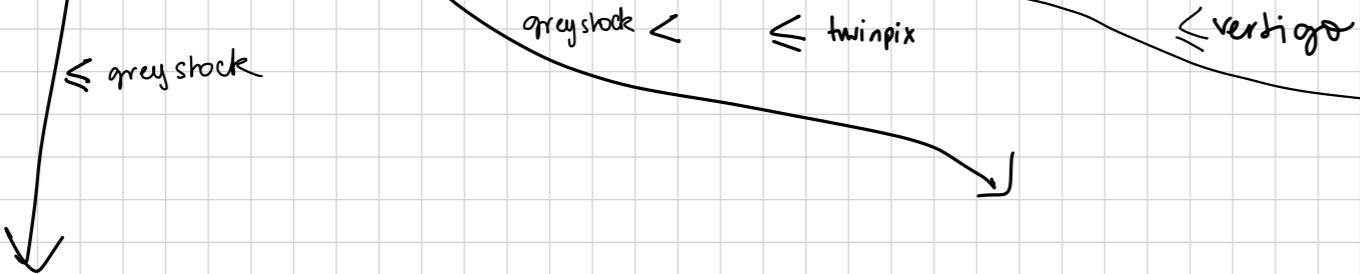
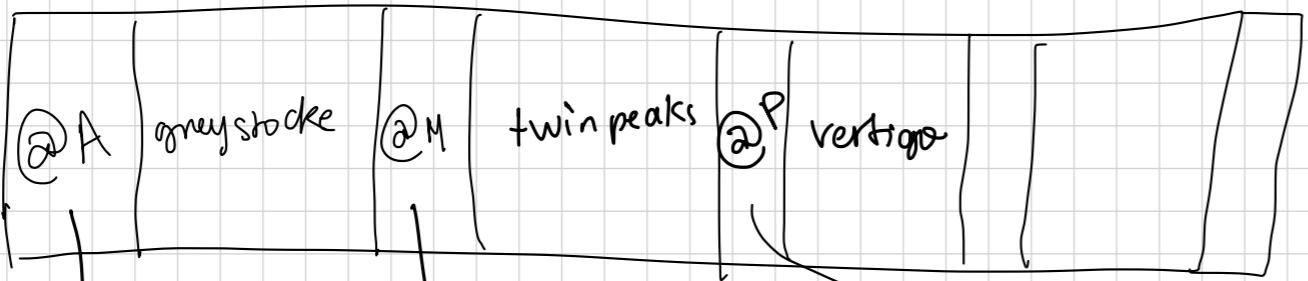


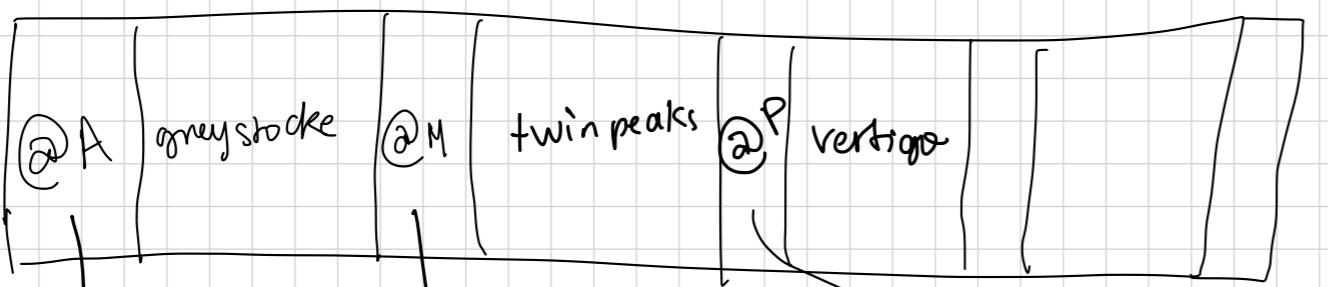
@M



Rq: fin le dernière de chapitre tableau entr celle qui est dans le racine

car ≤





≤ grey stocke

≤ twinpix

≤ vertigo

annie Hall 1977

@A

Brazil 1984

easy rider 1969

grey stocke 1984

@M

jurassic park 1992

psycho 1960

skinning 1980

twin Peaks 1990

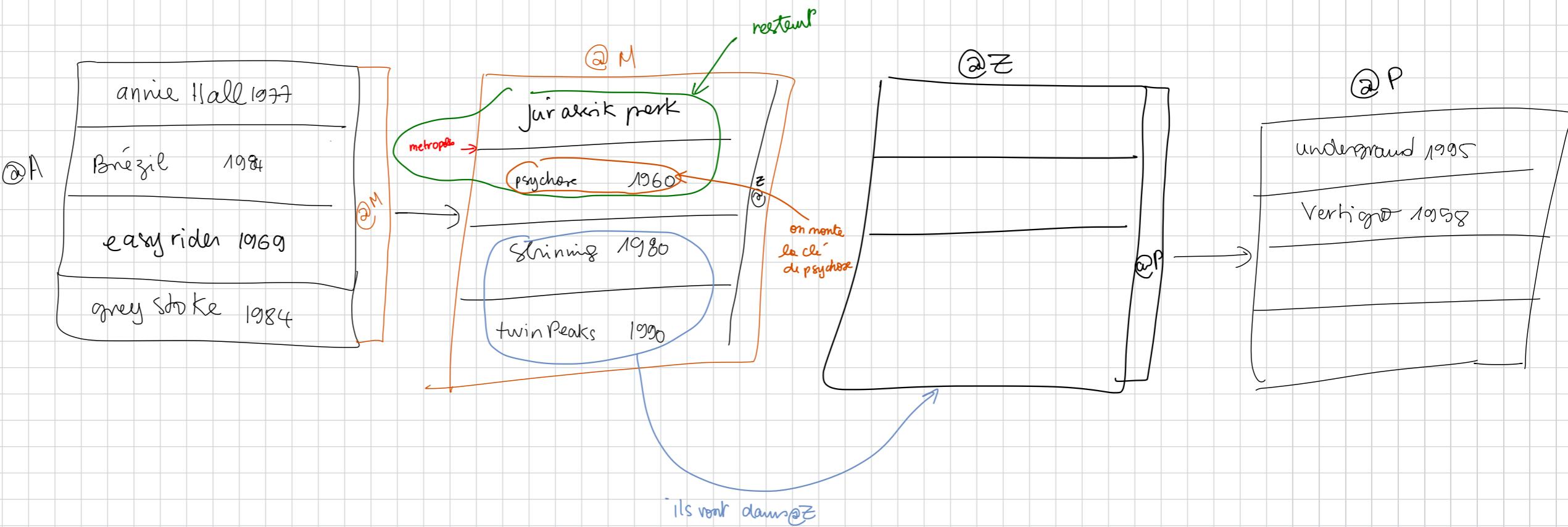
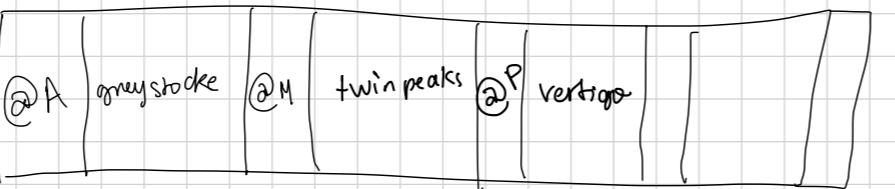
@M

@P

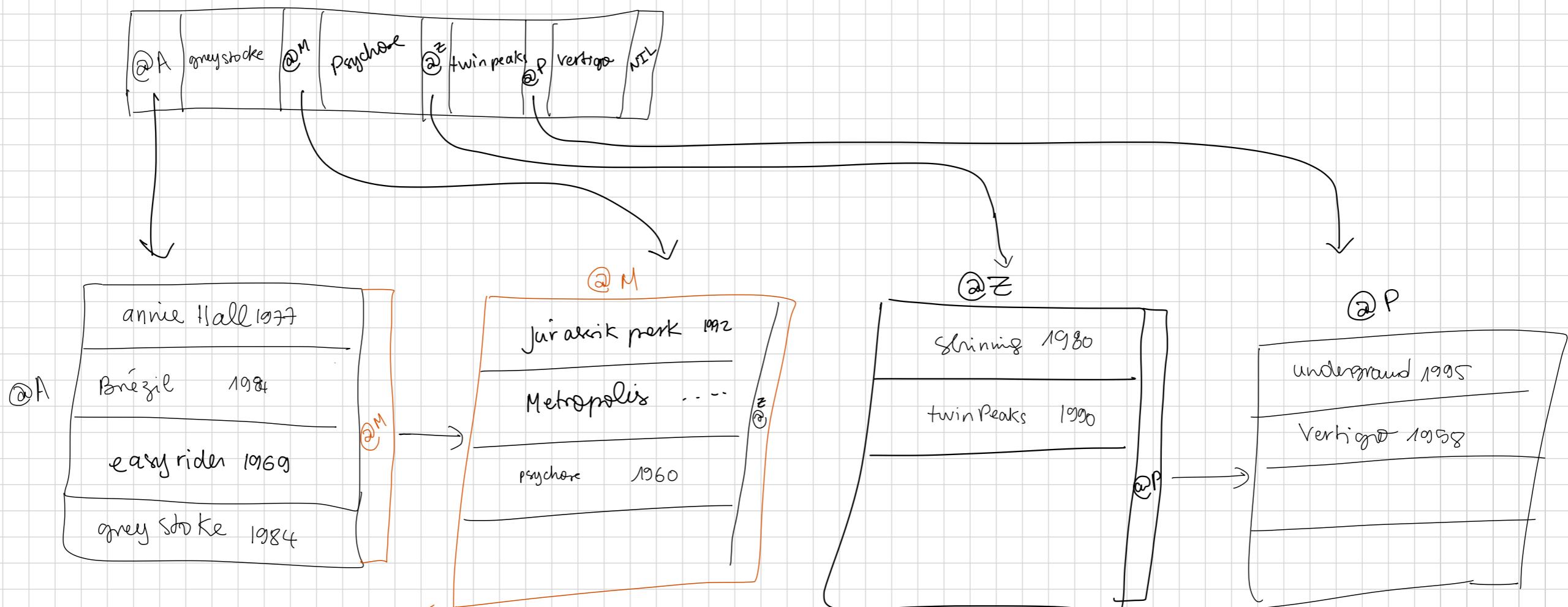
underground 1995

Vertigo 1958

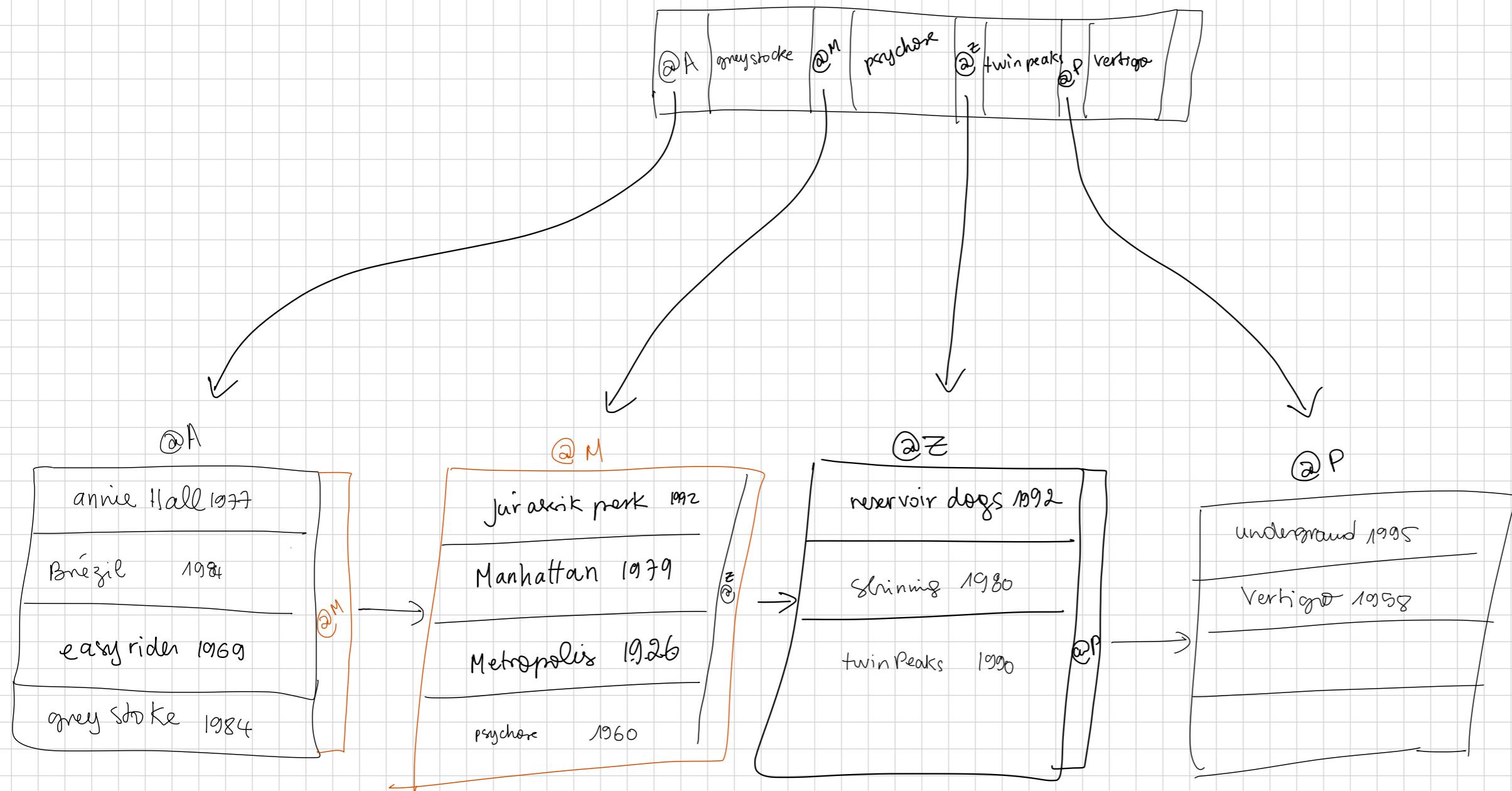
on ajoute métropolis  
pas de place



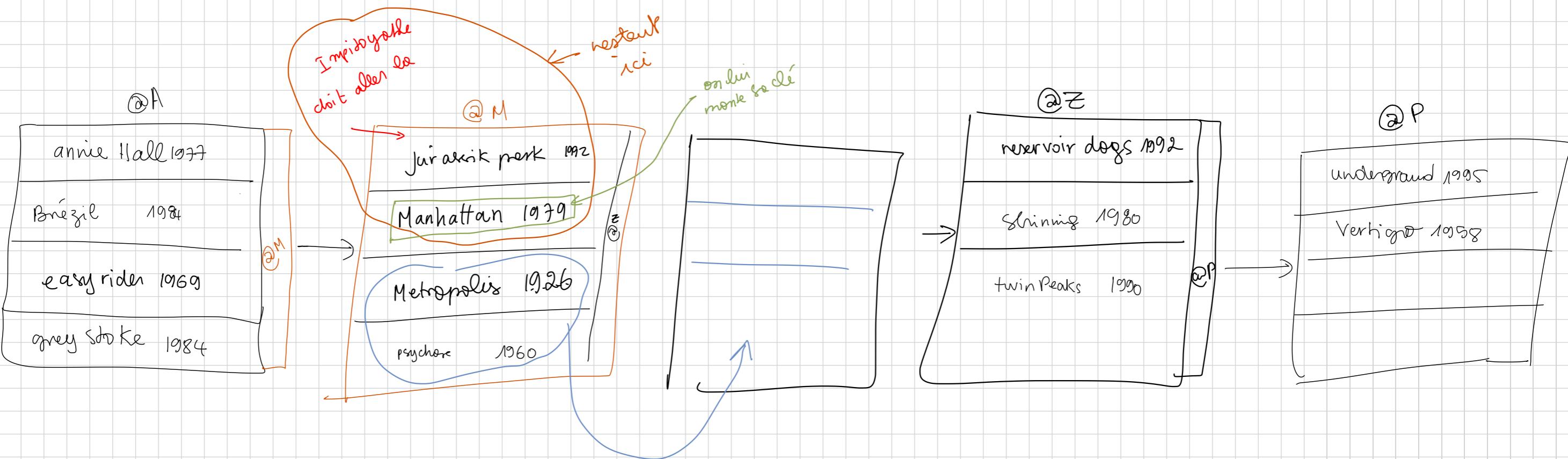
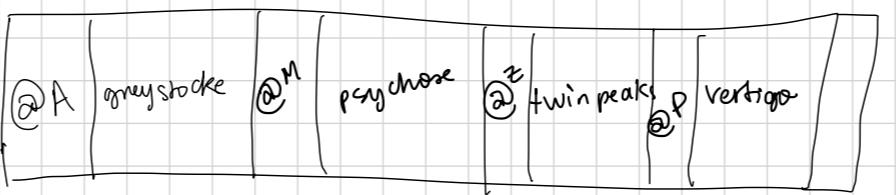
go dome go

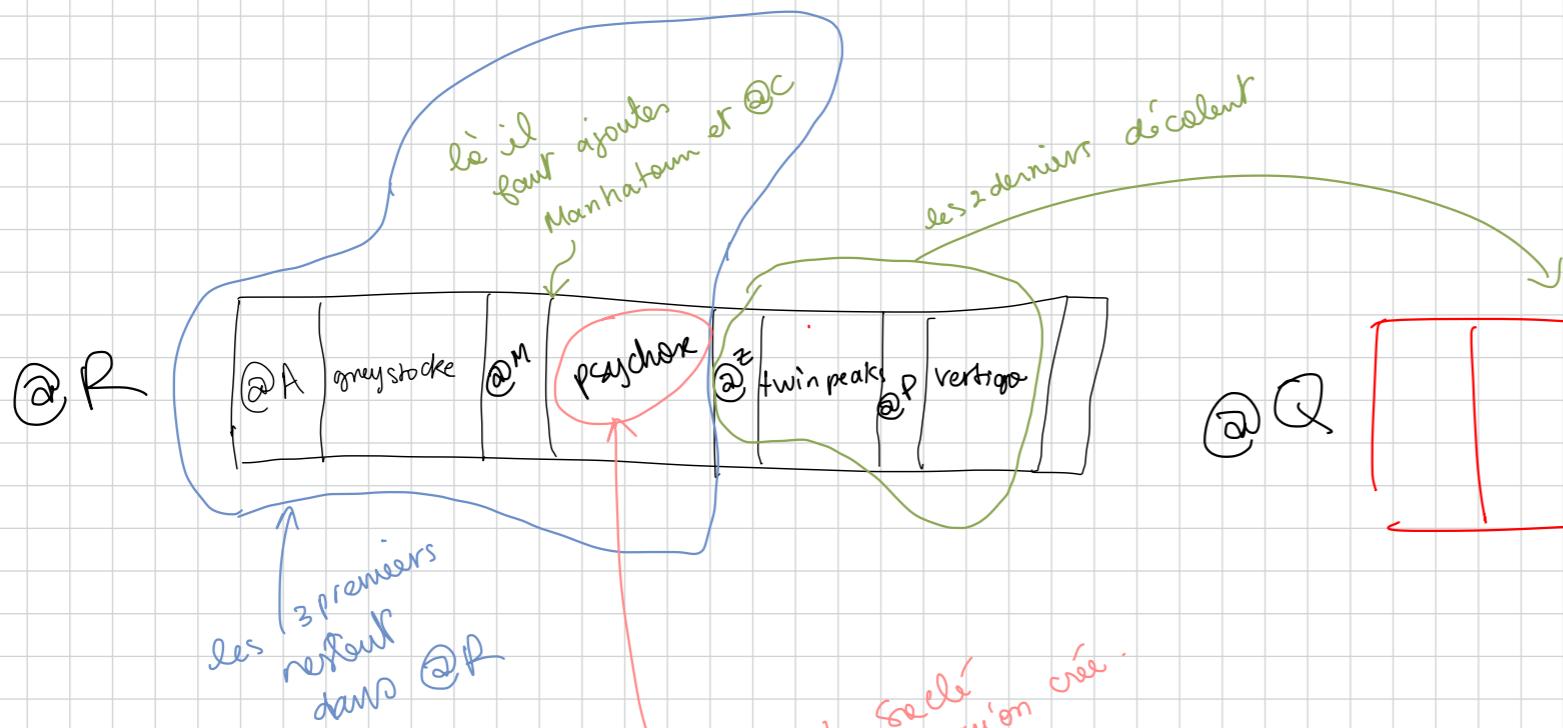


on insère Manhattan  
et reservoir dogs (place)

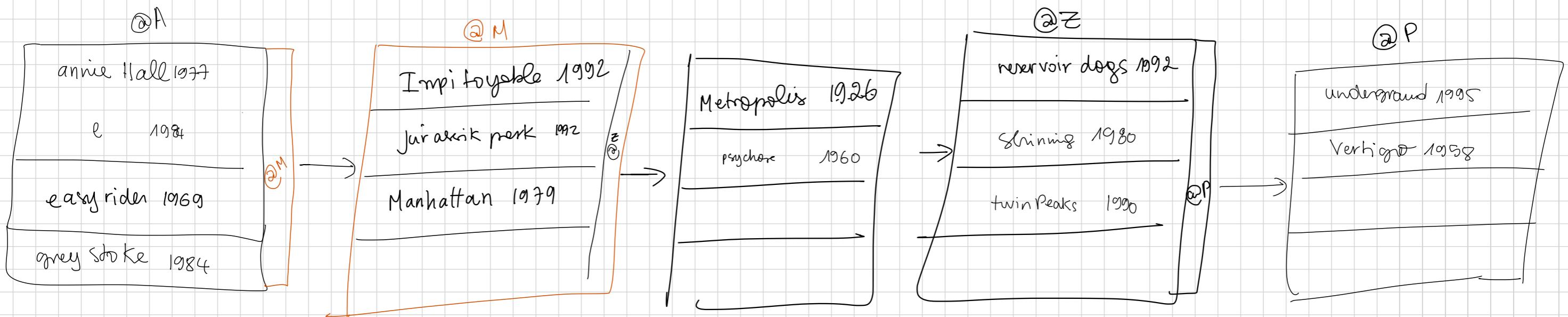


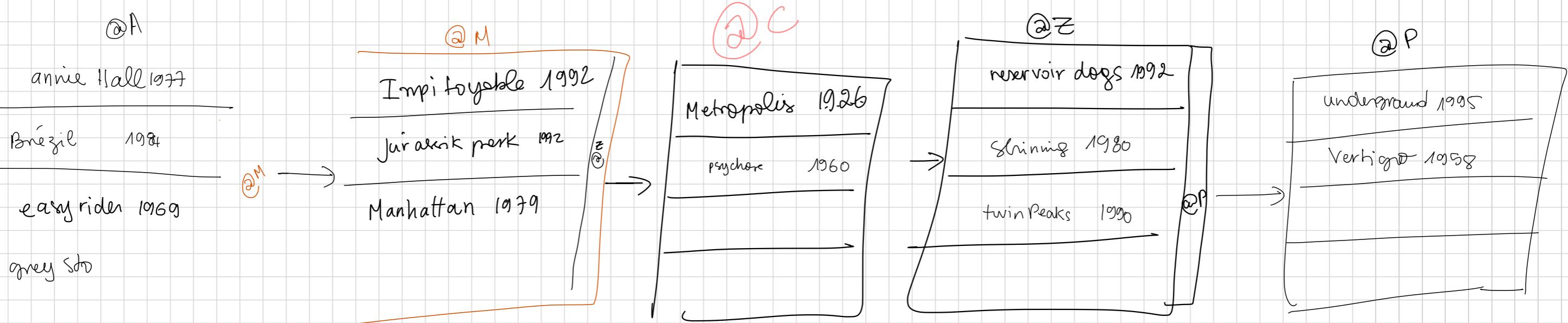
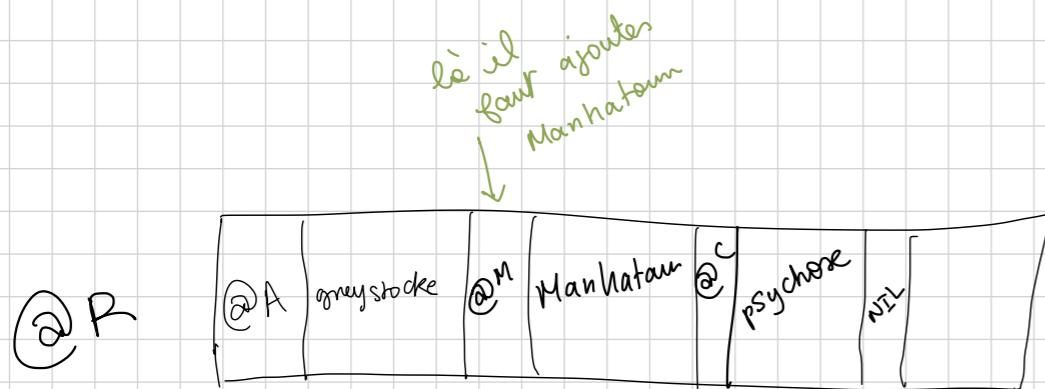
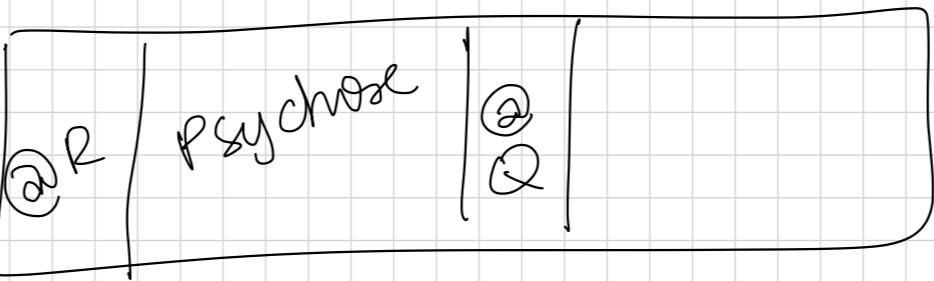
↓  
on ajoute I mitycèle





plus de place  
donc on allume  
une page





exercice 2 à rendre

Samedi Soir - | | |  
0 6 6



2.

?

Le nombre de niveaux d'un arbre et la taille d'un arbre sont deux concepts différents en informatique, qu'il s'agisse d'un arbre général ou d'un arbre binaire. Voici une explication des deux notions et comment les calculer :

#### 1. Nombre de niveaux (ou profondeur) d'un arbre :

Le nombre de niveaux d'un arbre (ou sa profondeur) correspond à la longueur du chemin du nœud racine au nœud le plus profond de l'arbre. Ce n'est donc pas la même chose que la taille.

- **Formule générale :** Le nombre de niveaux est calculé comme la **profondeur maximale** de l'arbre, c'est-à-dire le nombre maximum de relations parent-enfant du nœud racine à une feuille.
- Si l'arbre n'a qu'un seul nœud (la racine), alors le nombre de niveaux est 1.
- Pour un arbre avec plusieurs niveaux, il faut parcourir l'arbre et trouver la profondeur maximale des nœuds.

Dans un arbre binaire complet, le nombre de niveaux  $h$  est souvent exprimé en fonction du nombre total de nœuds  $n$  comme :

$$h = \log_2(n + 1)$$

Mais cette formule suppose que l'arbre est complet, c'est-à-dire que tous les niveaux sauf peut-être le dernier sont entièrement remplis.

Exercice à rendre : exercice 1.2

① Taille d'un nuplet = taille d'un enregistrement =  $7 + 50 + 50 + 20 + 23 = 150$  oct

② NTuples (client) = 300 000

$$\text{③ Bfactor (client)} = \left\lfloor \frac{\text{taille Page}}{\text{taille nuplet}} \right\rfloor = \left\lfloor \frac{3000}{150} \right\rfloor = 20 \text{ enregistrements/page}$$

④ Nblock (client) = le nombre de Pages nécessaire pour stocker tous les clients =  $\frac{300000}{20} = 15000$  Pages

⑤ Nlevels =  $\log_m N$   
 ↑  
 le nombre total de clés  
 le nombre d'adresses/page intérieure

$$m = \frac{\text{taille page}}{\text{taille clé relation} + \text{taille adresse}} = \frac{3000}{8+7} = 200 \text{ adresses/page}$$

donc  $N_{levels} = \log_{200} (300000) \approx 3$  et donc (2 niveaux index + 1 niveau enregistrement)

feuilles

⑥ a. le niveau des feuilles :  $\frac{\text{nbre de tuples}}{\text{Bfactor}} = \frac{300000}{20} = 15000$  pages pour le niveau feuilles

b. le niveau intérieur :  $\frac{\text{nombre de pages du niveau feuilles}}{m} = \frac{15000}{200} = 75$  Pages pour le niveau intérieur

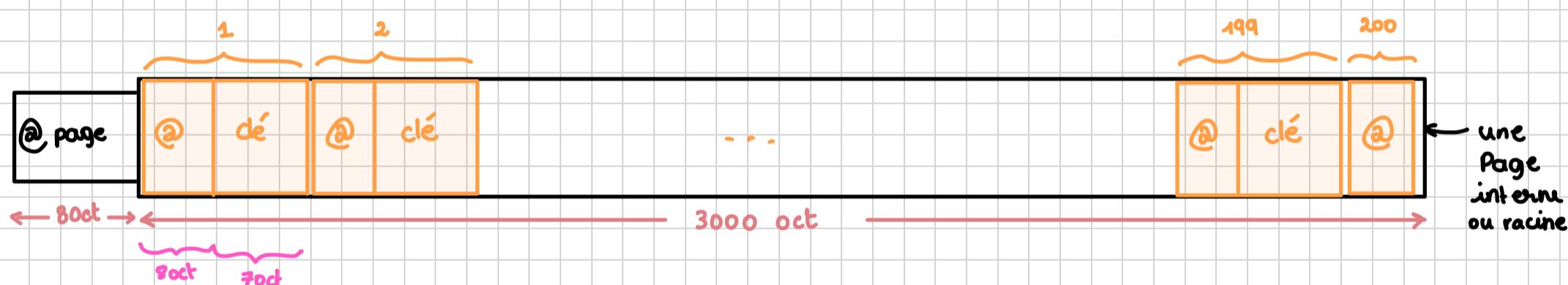
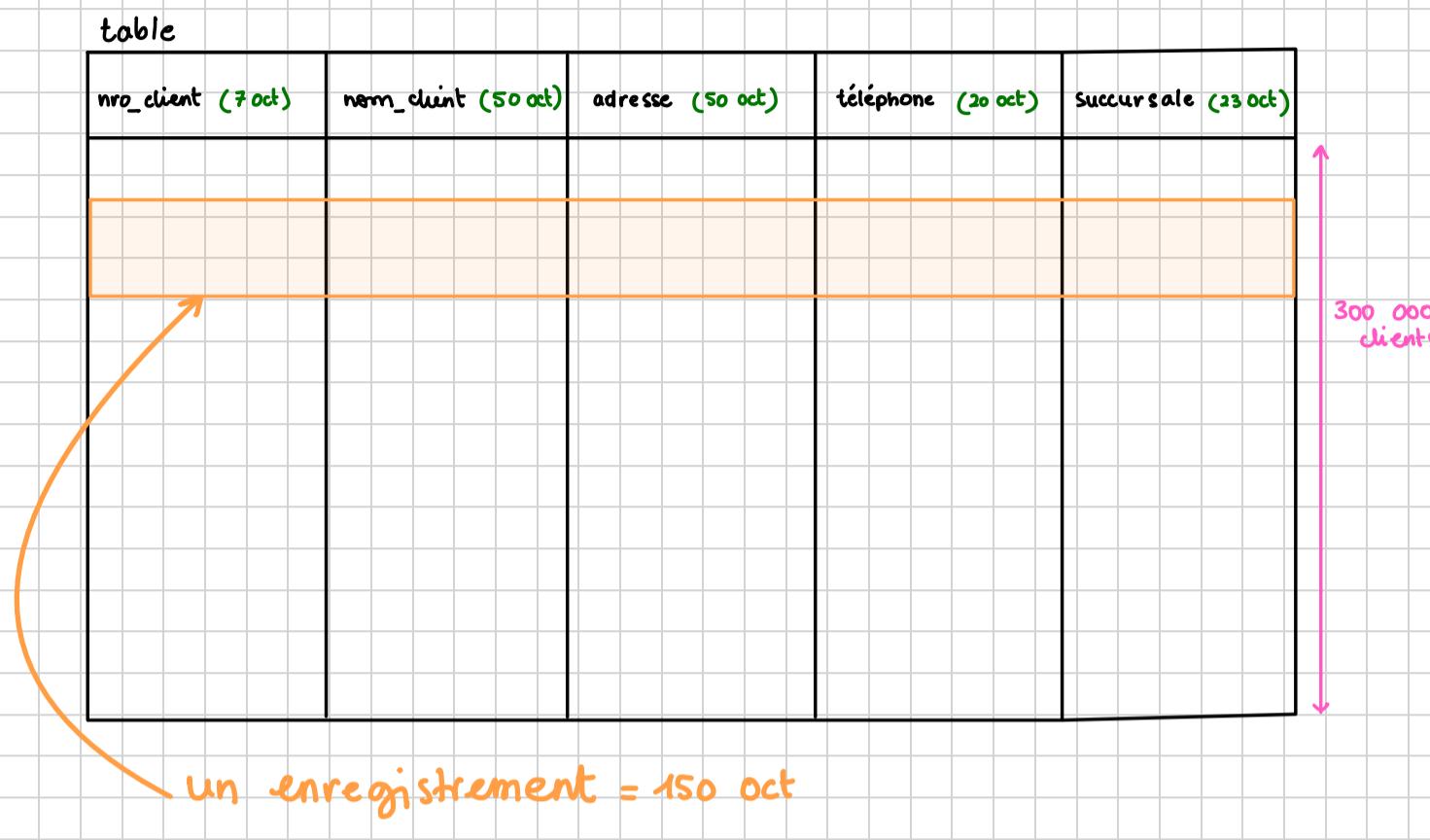
c. le niveau racine :  $\frac{\text{nombre de pages du niveau intérieur}}{m} = \left\lceil \frac{75}{200} \right\rceil = 1$  page pour la racine

Le cela confirme que le nombre de niveaux dans l'arbre miniaire est bien = 2 + 1 (feuilles)

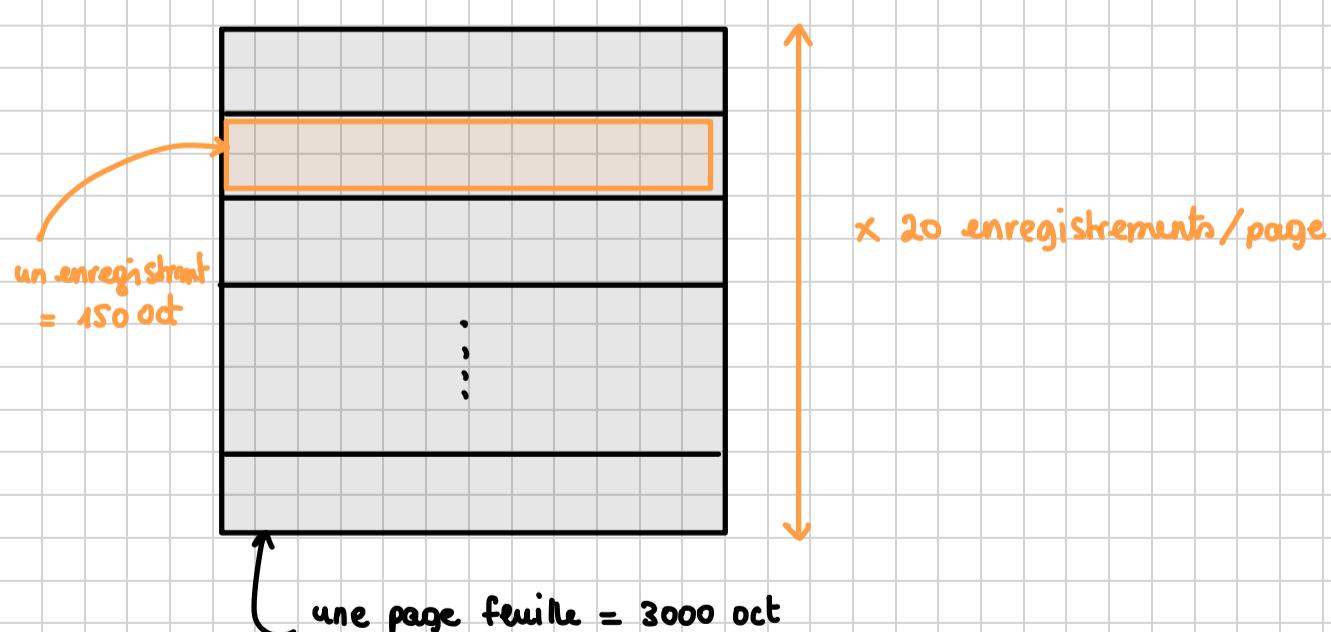
et donc  $N_{block\_nro\_client} = 15000 + 75 + 1 = 15076$  au total

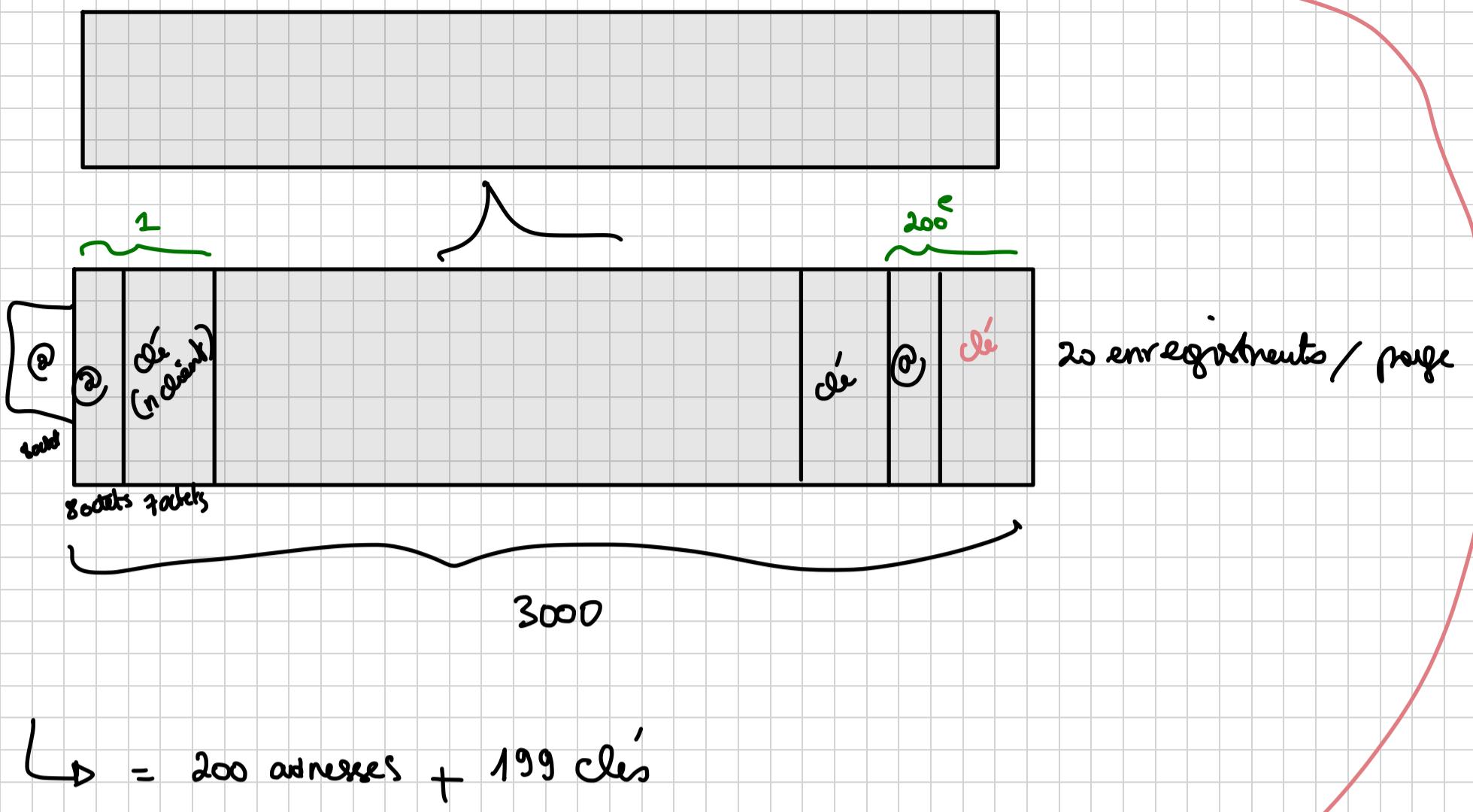
↑      ↑  
 niv feuilles      niv intérieur      racine de l'index  
 de l'index

## Schémas explicatifs



car la table est indexée par rapport au n\_client





N levels

Brailliom

# Solution de Hiba

TD 2exo 2

1) taille de ntuple  $\Sigma$  taille des attribut =  $7+50+50+20+23 = 150$  octet

2)  $N_{\text{tuple}}(\text{clint})$ : le nombre Totale des client = 300 000

3)  $B_{\text{Factor}}(\text{clint}) = \frac{\text{taille de la page}}{\text{taille d'un ntuple}} = \frac{300\ 000}{150} = 20$  tuples.

4)  $N_{\text{Bloc}}(\text{clint}) = \frac{\text{Nombre de client total}}{\text{Nombre de client par page}} = \frac{300\ 000}{15\ 000} = 15000$  bloc

5)  $N_{\text{Leaf}}_{\text{nro-clnt}}(I) =$

On a 15000 bloc pour représenter les feuilles, donc 15000 nœuds feuilles sur le niveau 0

La capacité d'un nœud =  $\frac{3000}{15} = 200$  clt donc 201 enfants

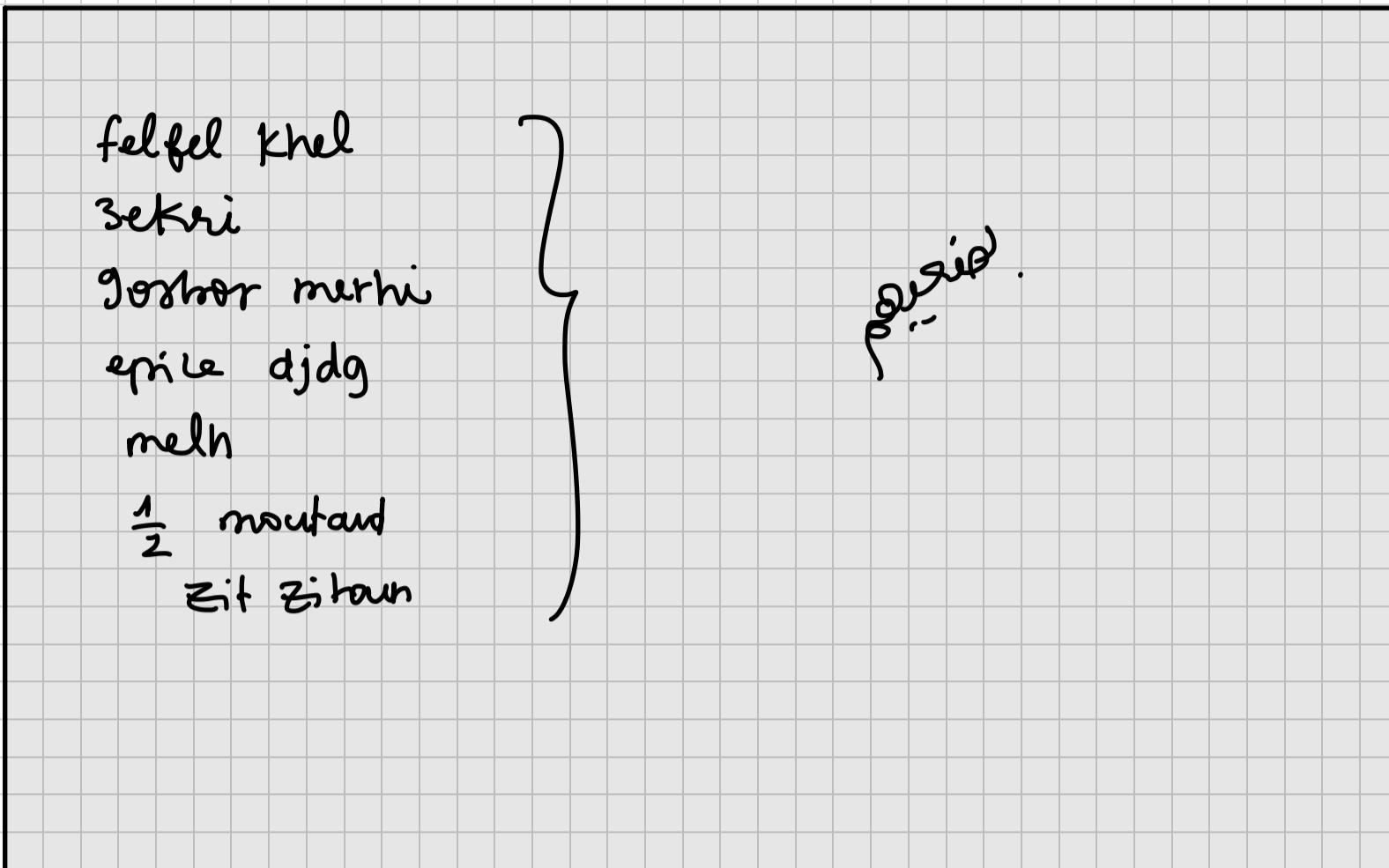
Nombre de nœud intérieur (niveau 1) =  $\left\lceil \frac{15\ 000}{200} \right\rceil = [75,63] = 76$  nœud.

Nombre de nœud intérieur (niveau 2) =  $\left\lceil \frac{76}{200} \right\rceil = 1$  c'est la racine.

donc l'arbre  $\boxed{\text{possède 3 niveaux}}$ :

c)  $N_{\text{Bloc}}_{\text{nro-clnt}}(I) = \sum_{i=0}^2 N_{\text{Bloc}}(\text{niveau}_i) = 15000 + 76 + 1 = 15076$

la solution de ChatGPT: à copier (bien expliquée)

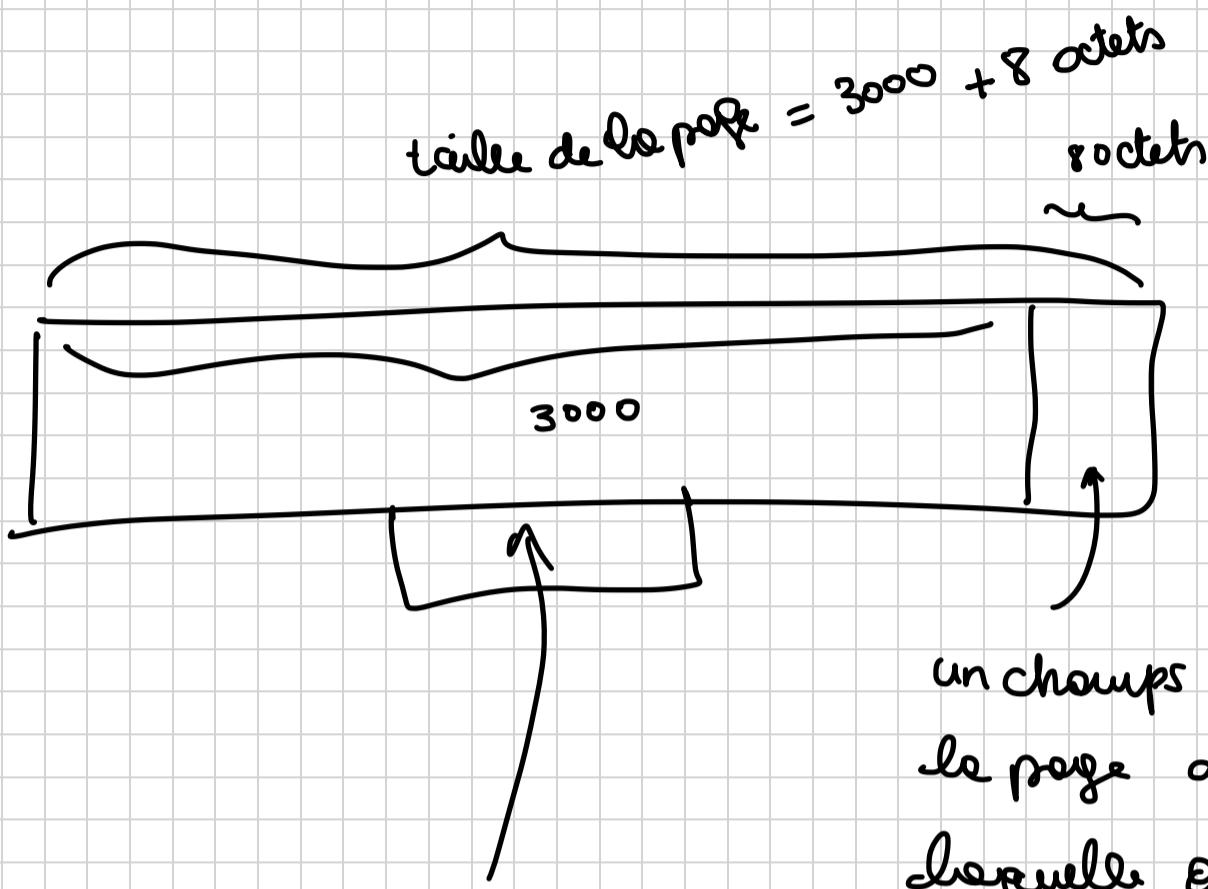


5min gechauffer  $160^{\circ}$  porcier cuire  
20 min djiha fougasse }  
20 min gelhiboum }  $160^{\circ}$   
10 min tahnir .

Connexion de la prof.

taille d'une page = 3008

Rq :



adresse de  
la page

un champs dans  
la page dans  
lequel on peut  
stocker une adresse  
qui a la taille  
d'une page



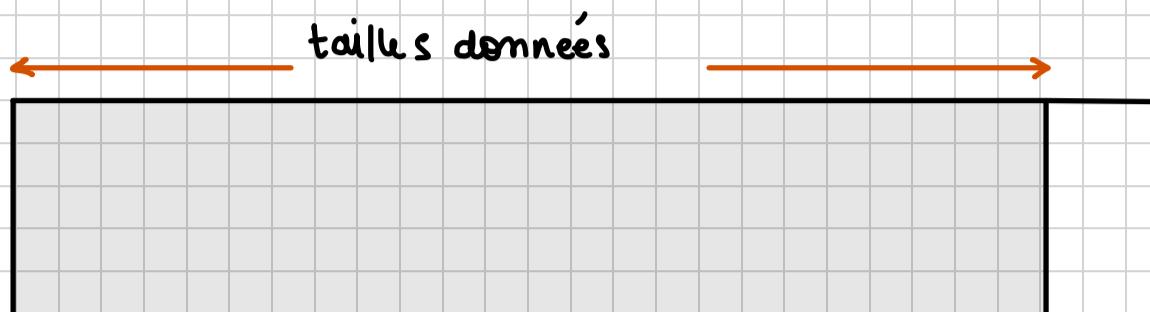
la liste  
chaînée par  
exemple .

\* les noeuds feuilles d'un arbre B<sup>+</sup> forment une liste  
chainée



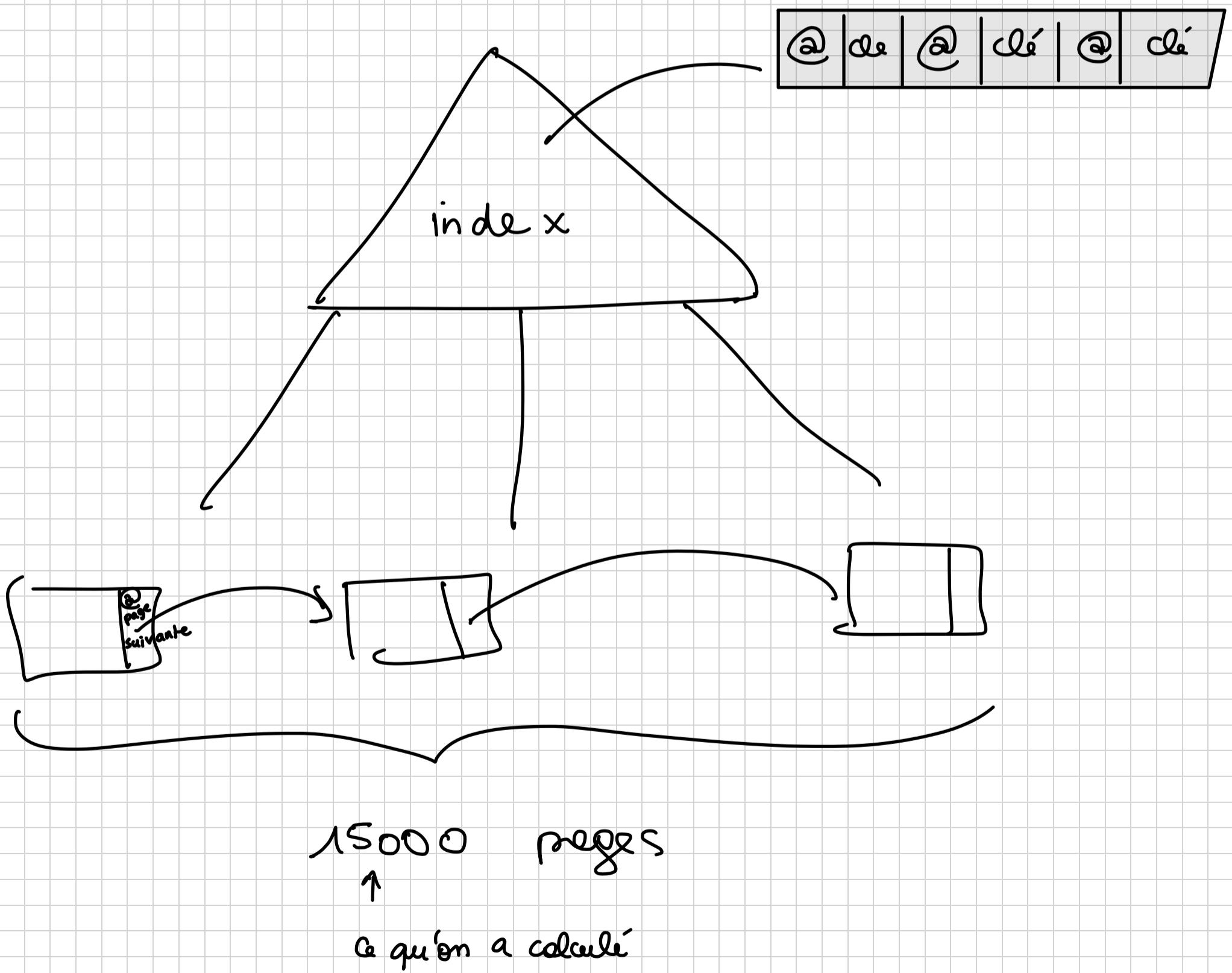
2.1 :

taille de la page = 3000 + 8



- Taille tuple =  $23 + 50 + 50 + 20 + 23 + 7 = 150 \text{ oct}$
- NTuples (client) = 300,000
- B factor (client) =  $3000 / 150 = 20$
- NB blocks (client) =  $300,000 / 20 = 15,000 \text{ blocs}$

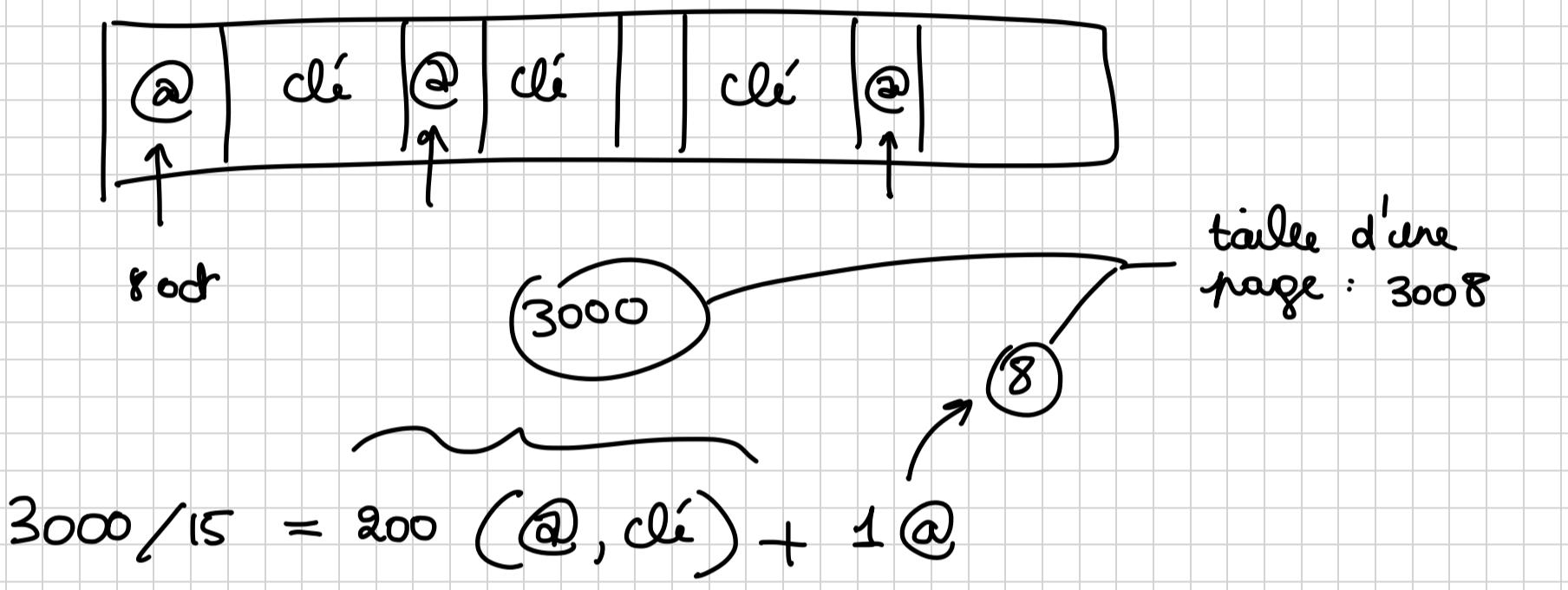
Rappel : on parle d'un arbre B+ qui



si c'était pas une liste chaînée, le 8 aurait été conservé dans les données aussi

Pour l'index :

on a une page = 3008 oct



$$= \underbrace{200 \text{ clés}}_{\text{pair}} \text{ et } 201 @$$

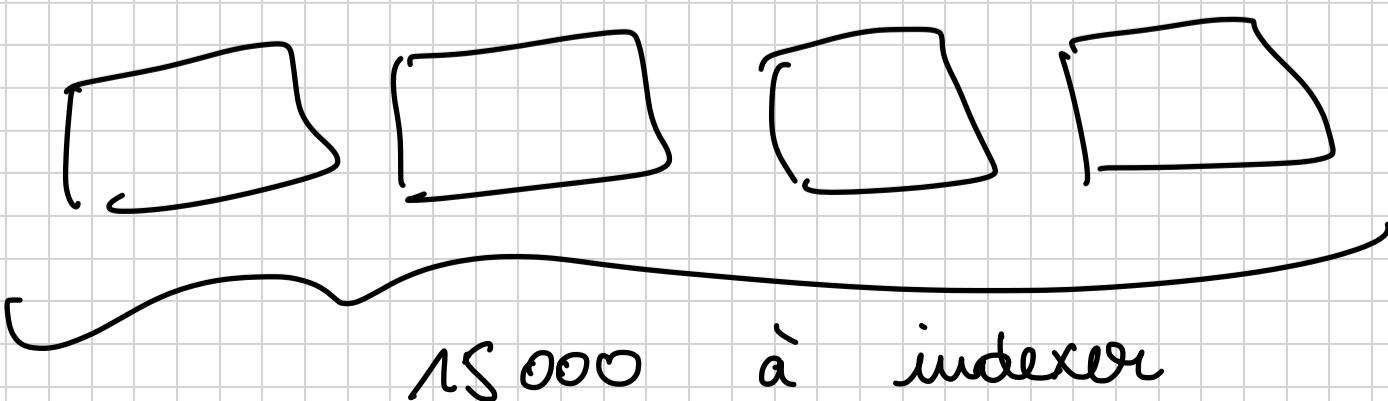
ici c'est pair  
les clés donc  
on peut l'utiliser

si c'était pas pair

on aurait enlevé  
une.

Car  $m \leq k \leq 2m$

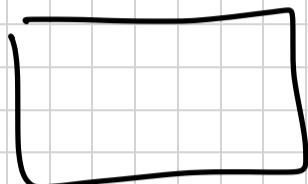
on a



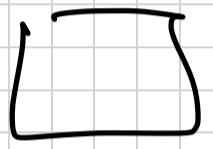
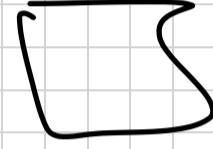
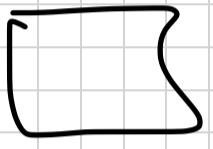
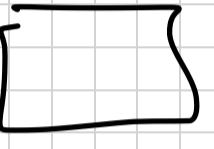
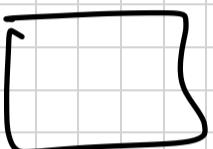
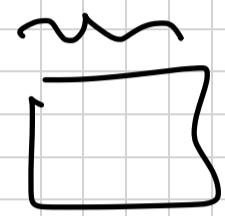
Sachant que chaque noeud peut indexer 201 pages

→  $15000 / 201 = 74$

peut indexer  
201

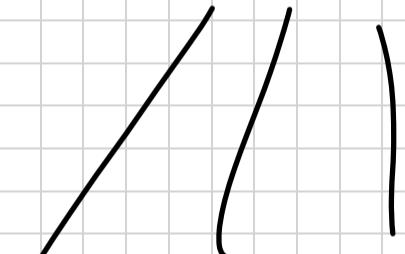


niv 1

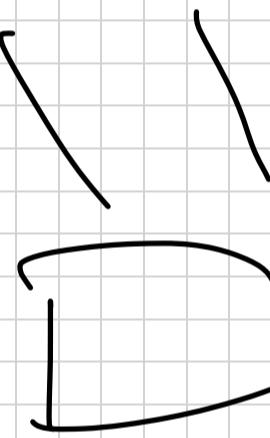


niv 2

74 pages (



- - - - -



niv 2

niv 3

15000

rien fait avec log

log 15000

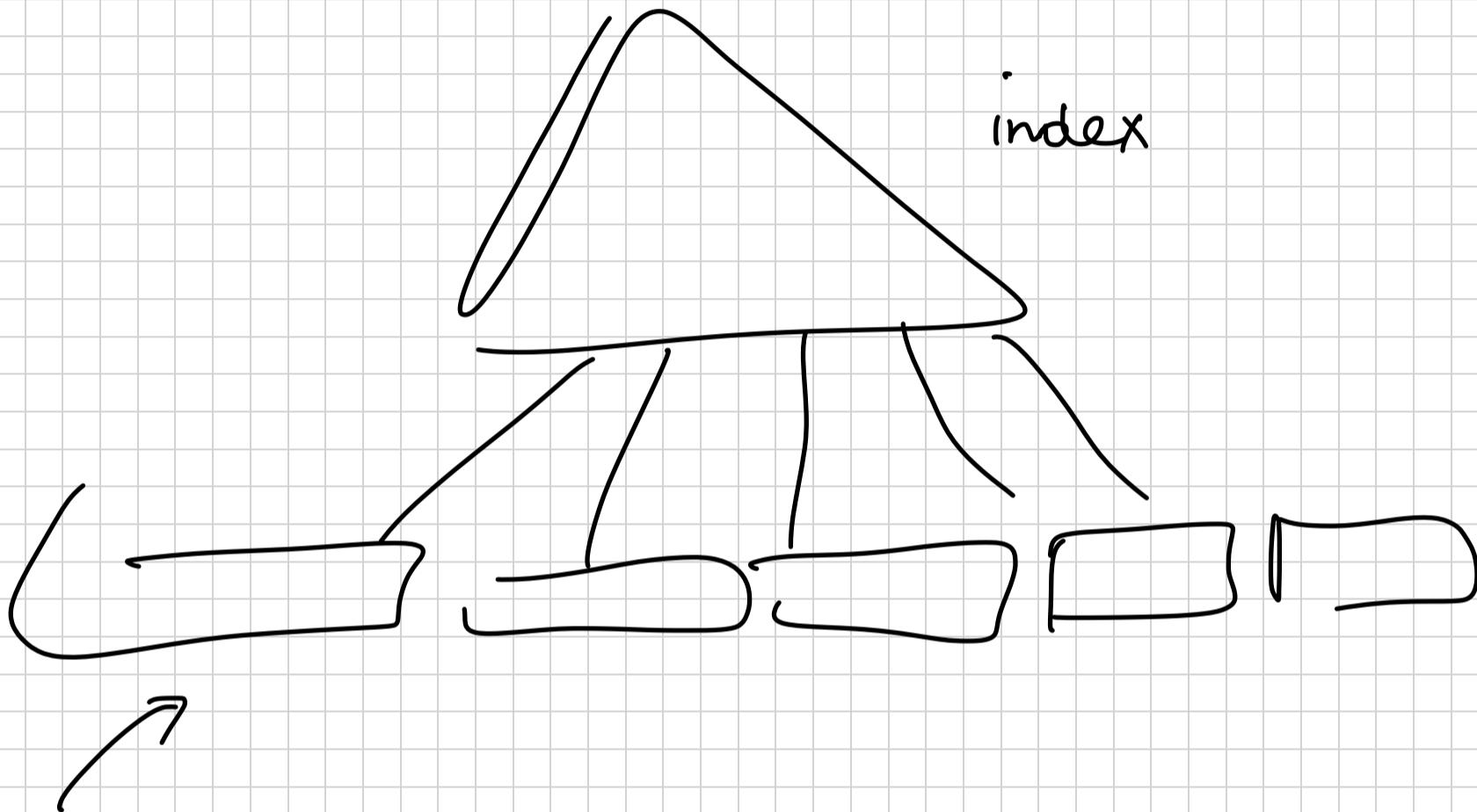
201

nombre d'adresses  
par page

le nombre de blocks pr stocker l'index =

$$74 + 1 = 75$$

NB block pour stocker la table = 75 + 15000



les données

3. on a un index <sup>secondaire</sup>  
~~dense~~  $\beta^+$  sur  
successive

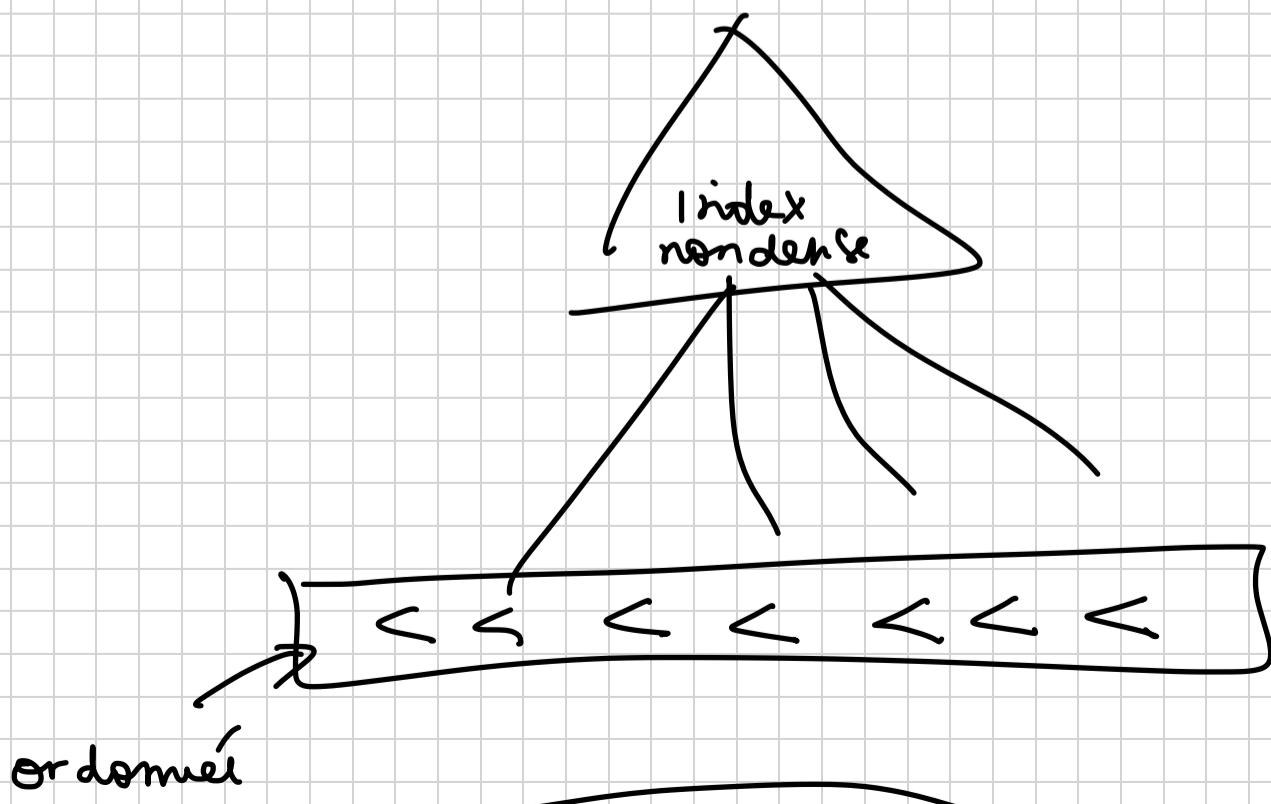


on peut pas avoir ici  
un index non dense.

car les données ne sont  
pas ordonnées par rapport  
aux successives (elles  
sont ordonnées par rapport à  $m^{\text{client}}$   
et pas successive)

dense : on indexe enregistrement par enregistrement

non dense : on indexe les pages et on peut le faire que  
si la liste chaînée est ordonnées



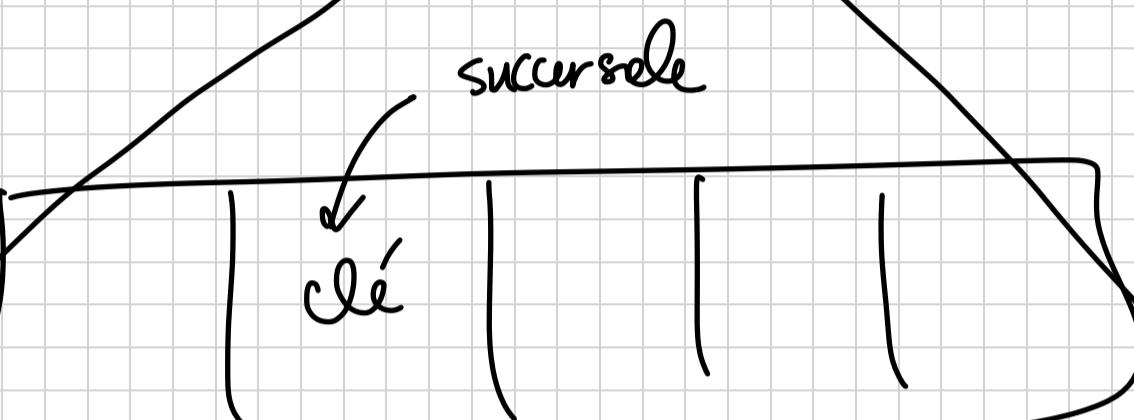
index primaire

(n clients, clé de recherche)

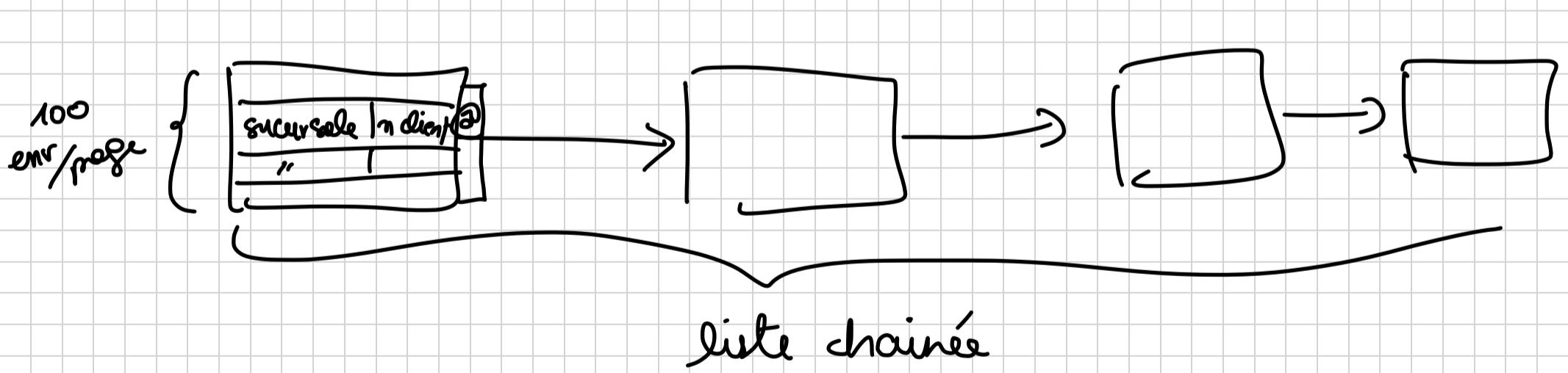
index secondaire

(succursale, n= client)

l'index II a :



index



liste chaînée

Rq: les enregistrements de l'index  $\neq$  les enregistrements de l'index

quelle est la taille des enregistrements

$$= (\text{successeur} + n \text{ client}) \\ 23 + 7 = 30 \text{ oct}$$

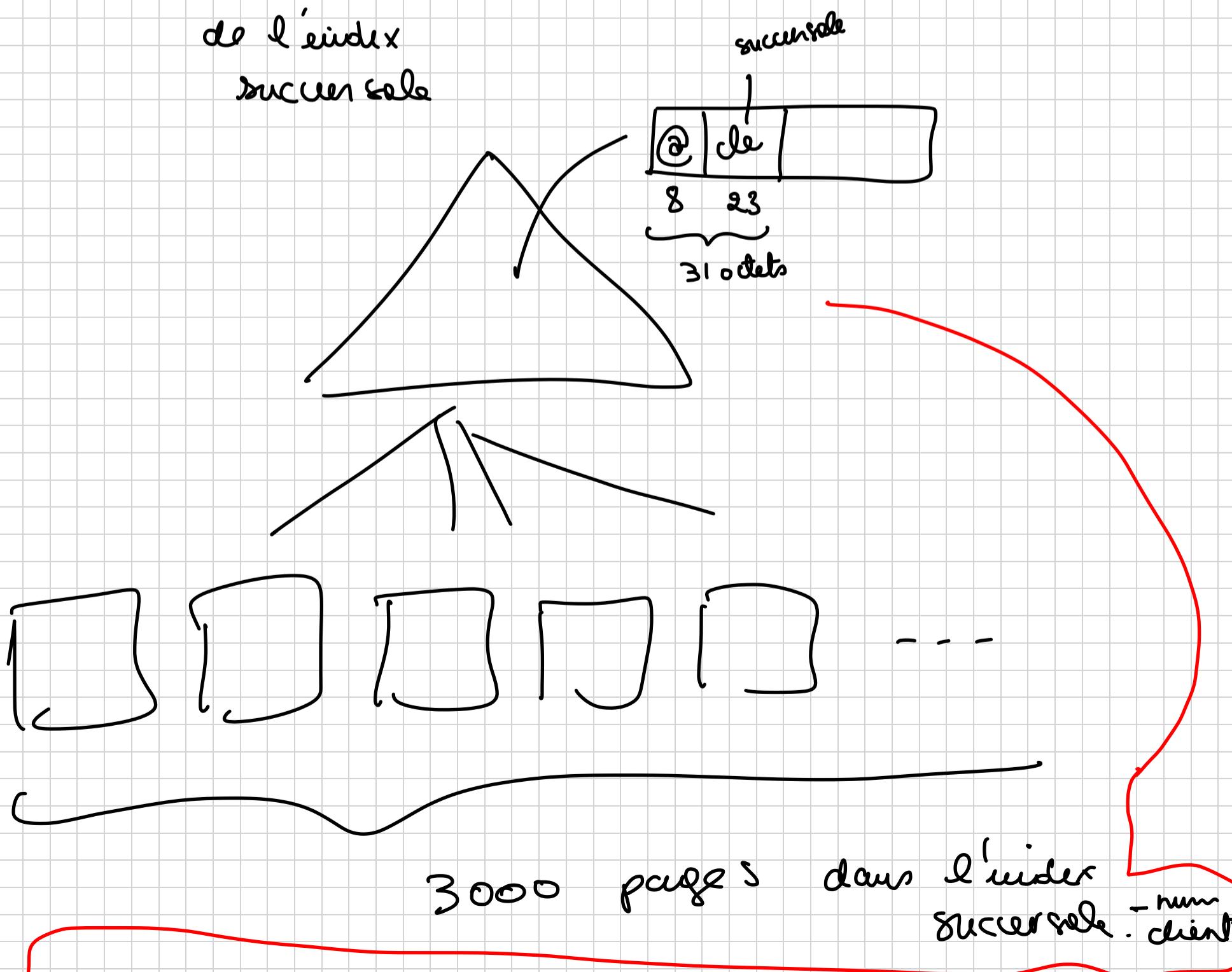
Nbre d'enregistrements par page :  $3000 / 30 = 100 \text{ enreg/ page}$

Combien de pages pour stocker les pages de l'index

↳ sachant qu'il ya un enregistrement pour chaque nuplet donc  $300000 \text{ Nuplet}$

$$\text{NB blocks (successeur)} = 3000000 / 100 = 30000 \text{ pages}$$

↑  
de l'index  
successeur



↓

Nb d'enregistrement (@, clé) =  $3000/31 = 96,17 = 96$

donc 96 clé et 97 @

pas donc

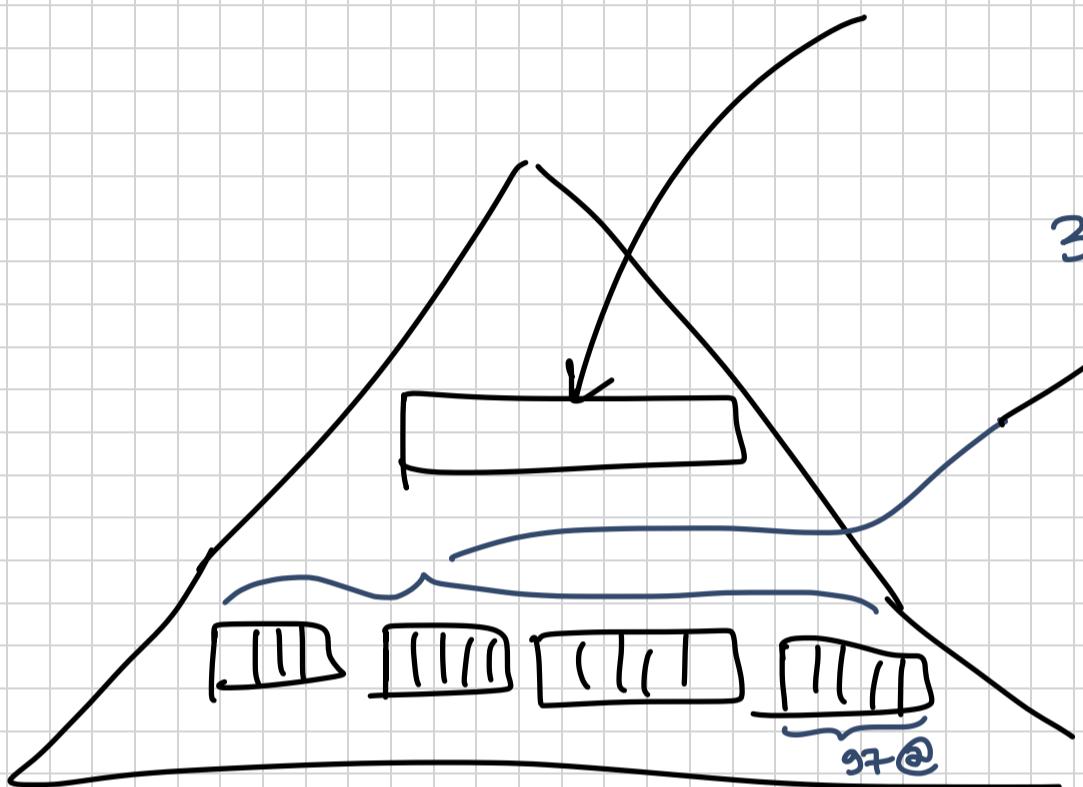
OK

1 seule page car

$$31 < 97$$

donc

$$3000/97 = 30,92 = 31$$



on aurait pu avoir comme index

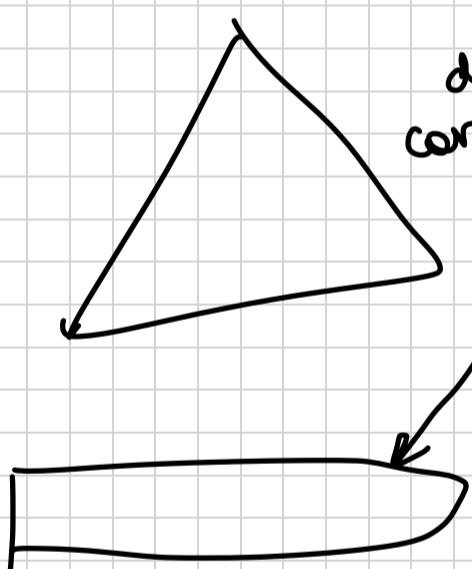


ce c'est la taille qui change.

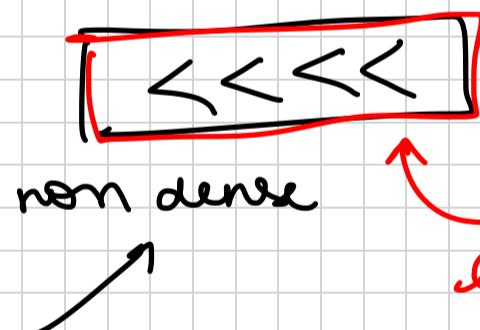
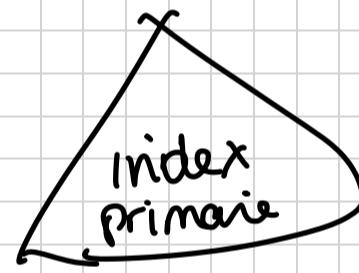
mais là en particulier, il ya que  
10 succursales, l'index ne peut pas  
avoir plus de 10,

on a donc pas besoin d'indexer chaque

l'index  
secondaire



l'index primaire



c'est c'est  
la table comme  
elle est stockée dans  
comment  
les données  
sont stockées

(une liste chaînée)

3.

$$SC \text{ succursale (client)} = 300.000 / 10 = 30.000$$

comme il ya 30.000 clients dans une succursale

donc il ya 30.000 client par succursale

ce vent dire que si on fait cette

greqeuette :

Select \*  
From client  
Where Succursale = 'Valeur'  
Résultat une table avec  
30.000 tuples.

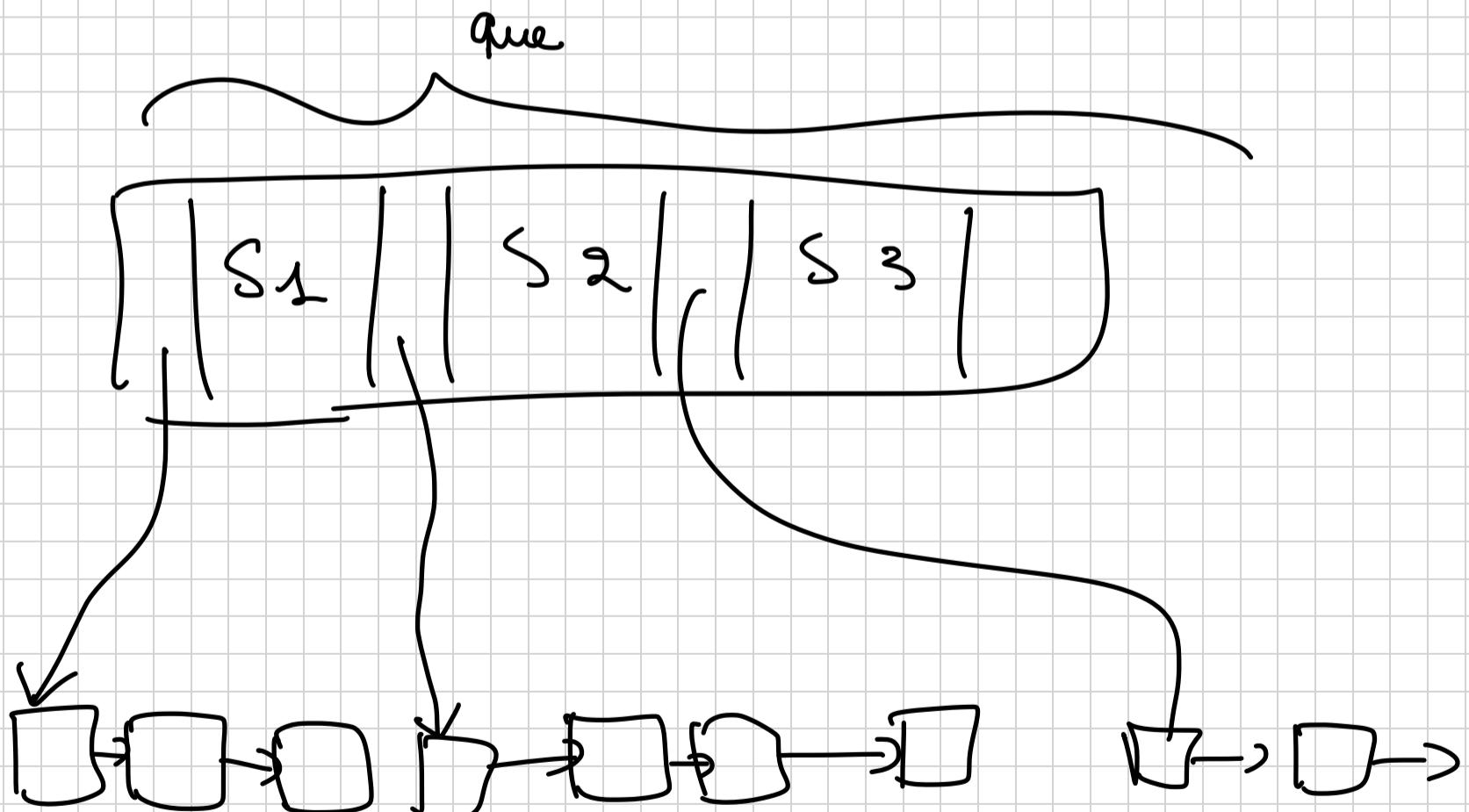
3. Supposons que pour la table client il y a un B+ associé à l'attribut succursale. Les clients sont distribués uniformément. Décrivez l'index secondaire en indiquant les éléments suivants :  
•  $SC_{succursale}(\text{Client}) = 3000$  tuples.  
•  $N\text{Distinct}_{succursale}(\text{Client})$   
•  $N\text{Level}_{succursale}(\text{I})$   
•  $N\text{Blocks}_{succursale}(\text{I})$

4. Soit un fichier tel que chaque page peut contenir jusqu'à 100 tuples d'index (un seul), et on suppose qu'un tuple d'index occupe 100 octets. Si le nombre d'articles, donnez la fonction de calcul de la taille de l'index pour les structures suivantes.

$$N\text{Distinct}(\text{client}) = 10$$

$$N\text{Level}_{succursale}(\text{I}) \approx (1 \text{ index} + 1 \text{ données})$$

$$N\text{Block}_{succursale} = 1 \quad \text{Avec} \quad \text{l'histoire du IO}$$



Profondeur = que l'index. ~~\* A~~

Nbr de bloc de l'index = 1

Nbr de bloc de racine = 3001 =  $\underbrace{3000}_{\uparrow} + \underbrace{1}_{\uparrow}$

nbre d'index  
de pages q  
(les feuilles) contient  
que la racine

4 • 10 article = 10 enregistrement / page .

ordonné → index non dense → on a une  
page .

Contrôle :

stockage / indexation

1 page = 10 articles

= 100 paires (clé, pointeur) ← nombre des  
enregistrements  
de l'index  
↑

l'index c'est un seul  
niveau

soit

on calcule en fonction de  $n$  le nombre de pages

$$1) \text{ n pages}(n) = \frac{n}{10} + \frac{n}{100} = \frac{10n + n}{100} = \frac{11n}{100}$$

le nombre  
d'articles

si on indexe  
par enregistrement car index dense

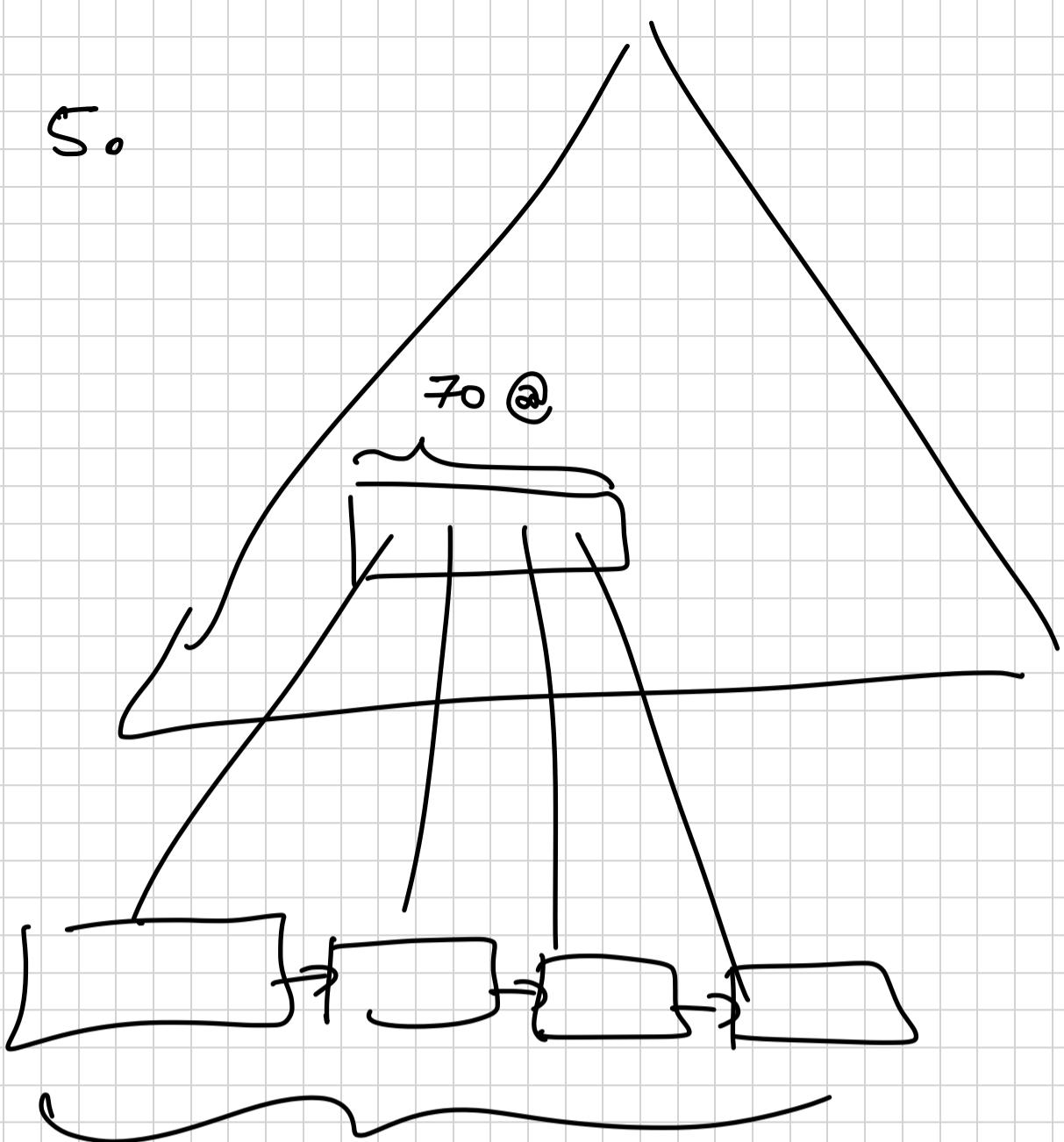
$$2) N \text{ pages } (n) = \frac{n}{10} + \frac{2}{100}$$

on indexe par page

car non-dense

↑  
car ordonnée.

5.

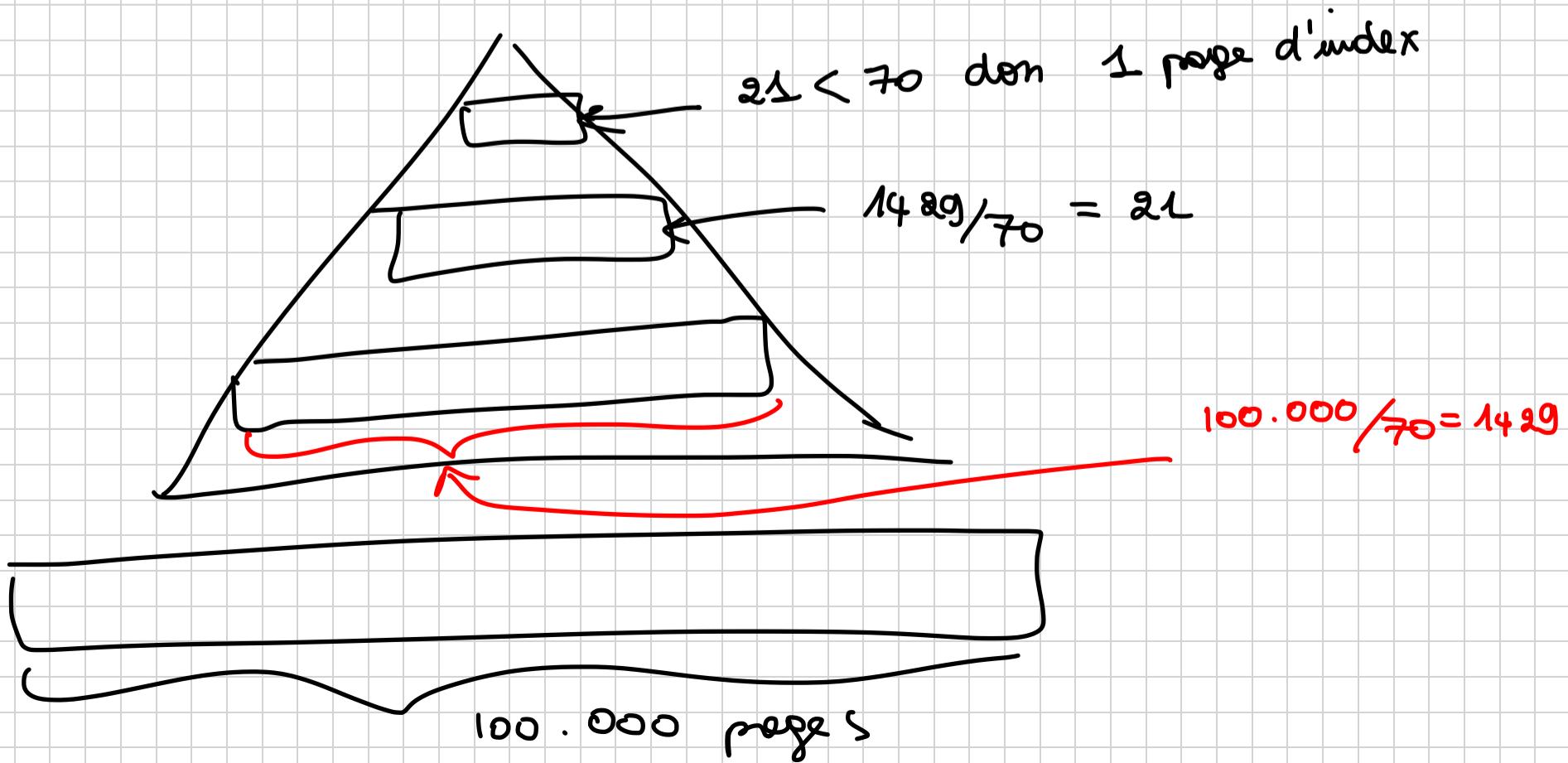


on veut savoir combien de pages dans la liste chaînée = 100 . 000 pages à indexer

$$n = \text{nbr d'articles} = 1m \text{ articles}$$

on a 10 articles par page

$$\text{donc nbr de feuilles} = \frac{1m}{10} = 100 . 000$$



a.

donc l'index primaire = 3 niv + niv données

nbre de pages = 100.000 + 1429 + 21 + 1

N level (F) 3 + des données

b. meilleur des ces

ReVV

meilleur des ces .



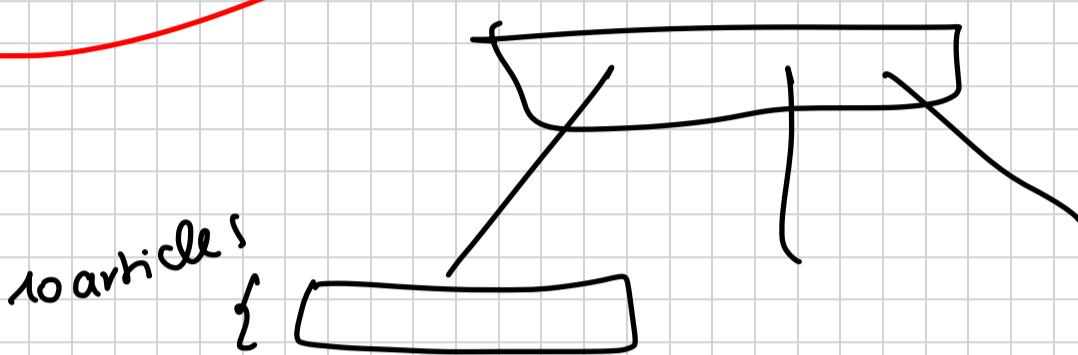
Rappel : algorithme de recherche dans un arbre bt

algorithme de recherche par rapport à une clé



on parcours, on trouve le clé  
le racine

• on va à la grille on trouve la clé et on descend à l'élément vers lequel elle pointe fermé dernier au node feuille

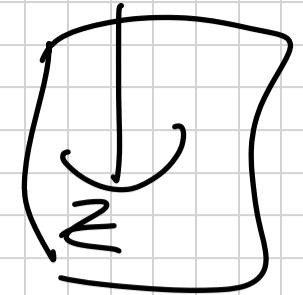
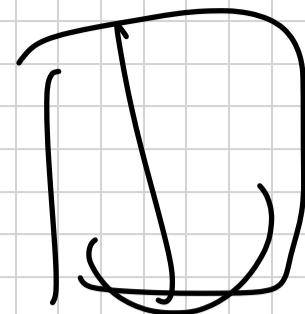
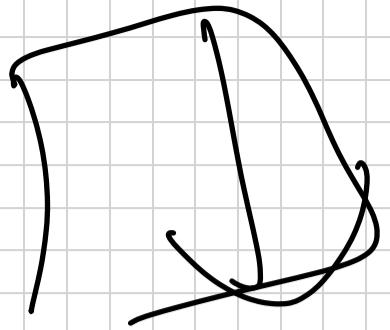
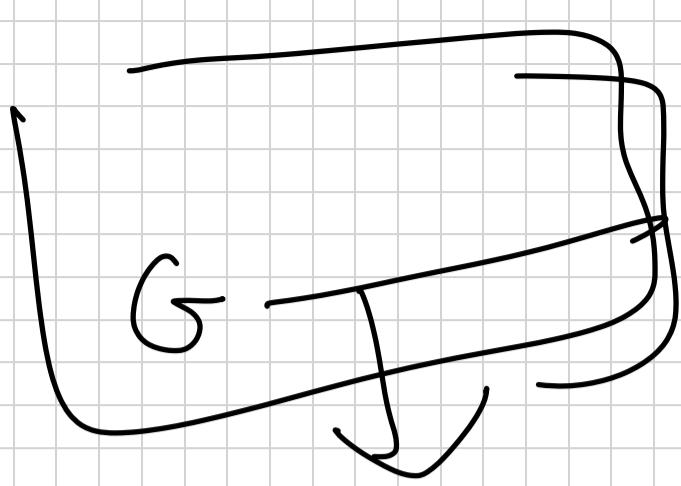


B<sup>+</sup>: c'est tjs ordonné :

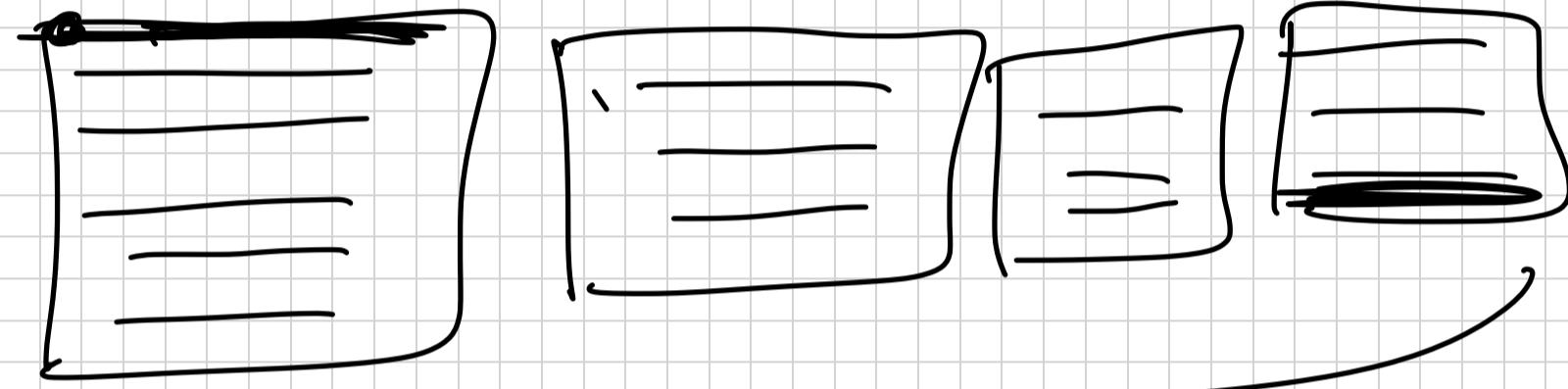
dans le meilleur des cas : les

pour récupérer 1000 articles on doit lire 100 pages + 3

dans le pire des cas : 101 pages + 3

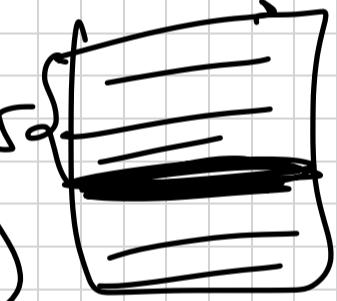
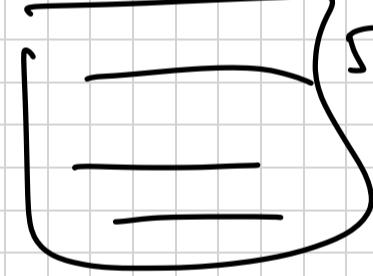
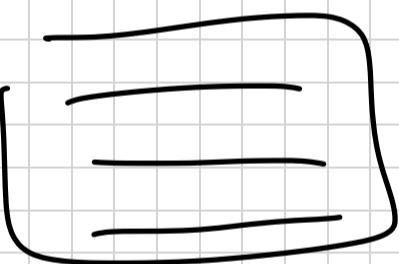
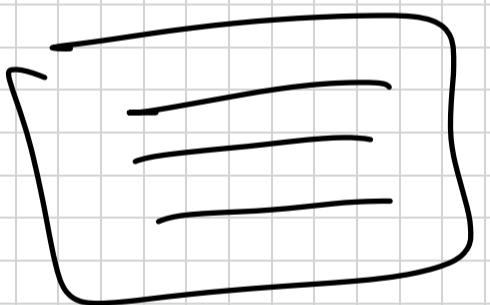


+1



meilleur  
des cas

100 pages



100

+ 1  
pire des  
cas