

# Faster and smaller Inverted Index



1

# Introduction

Contextualisation

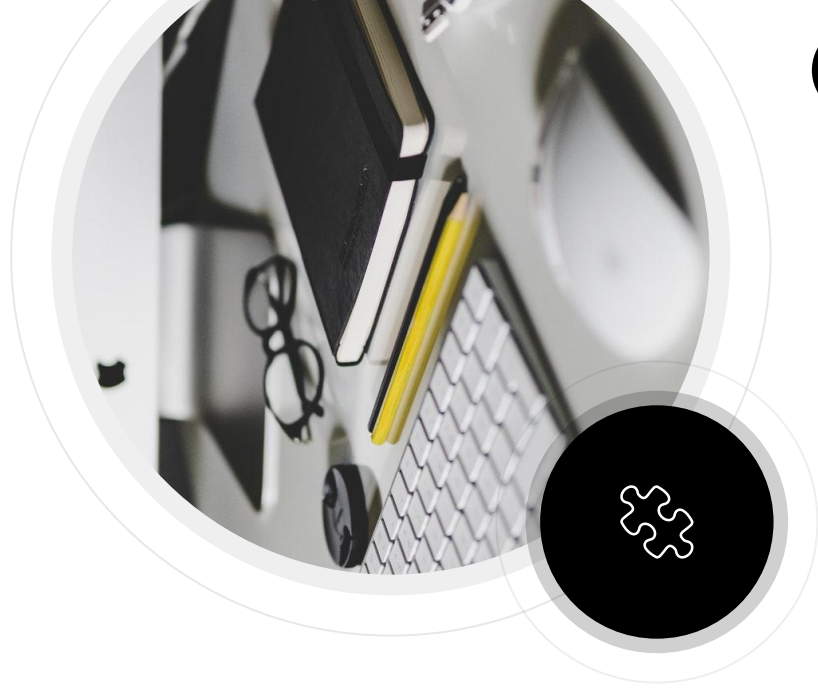
# Deux challenges

## Taille des données

Les moteurs de recherche ont de plus en plus de données à trier.

## Précision

L'utilisateur du moteur de recherche attend un résultat précis.



# En recevant une requête

- Filtration de documents intéressants
- Calcul de scores
- Tri et envoi des résultats

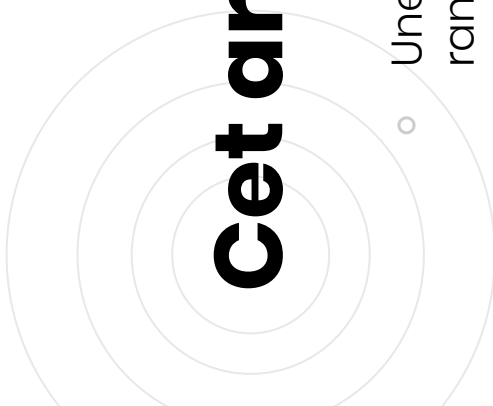
Dans les modèles classiques,

- Intersection booléenne
- Calcul de scores

Ou

- Ranked union (en utilisant les scores)

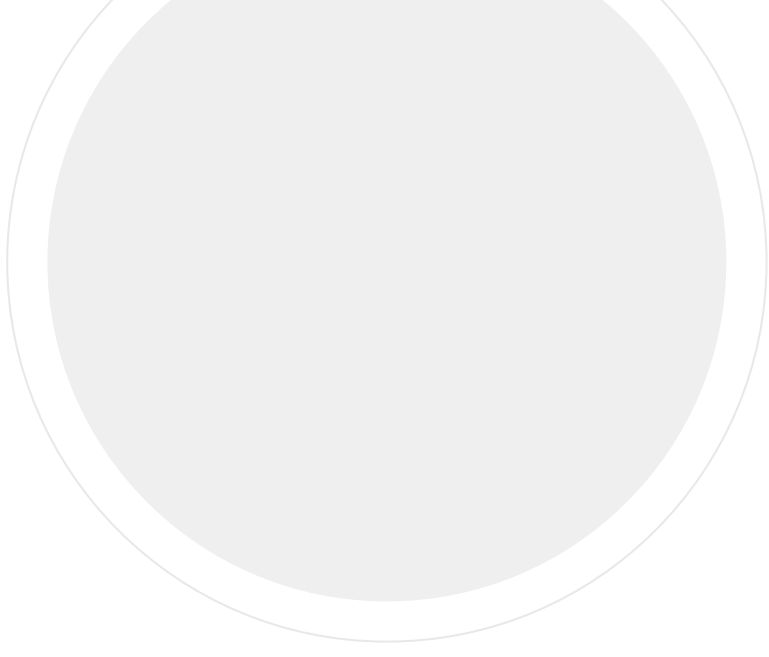




# Cet article propose

- Une méthode de calcul direct de ranked intersection
- Un index inversé plus condensé
- Une recherche plus rapide

À l'aide d'une structure de données non exploitées jusque là: **les treaps**





2

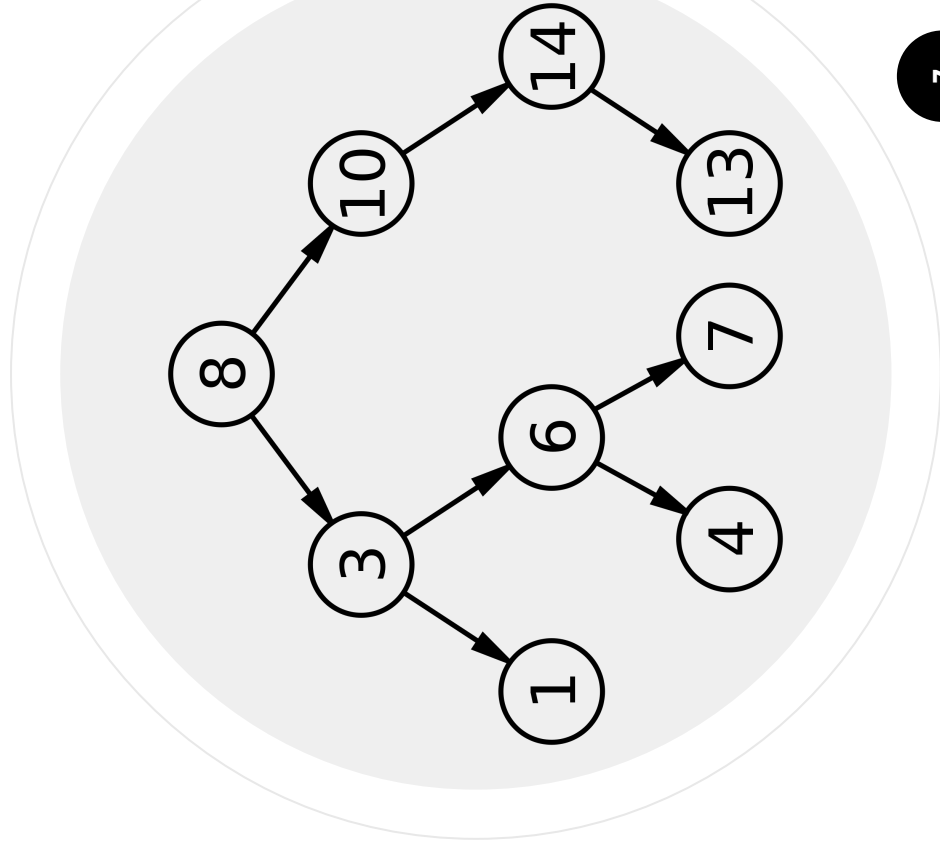
# Rappels

Brève introduction des concepts

# Binary search tree

La valeur de chaque noeud est plus grande que celle de l'enfant de gauche, plus petite que celle de l'enfant de droite.

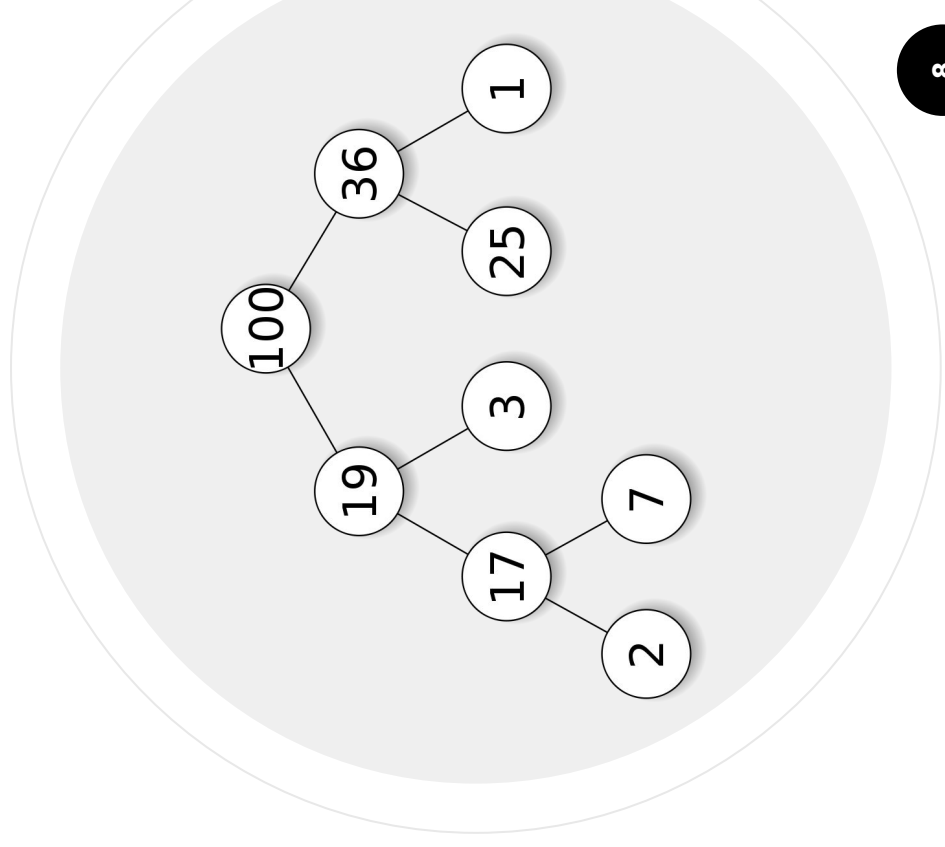
Traversée de gauche à droite



# Min Heap

La valeur de chaque noeud est plus grande que celles de ses enfants.

Traversée de haut en bas





# Treaps

Et leur représentation

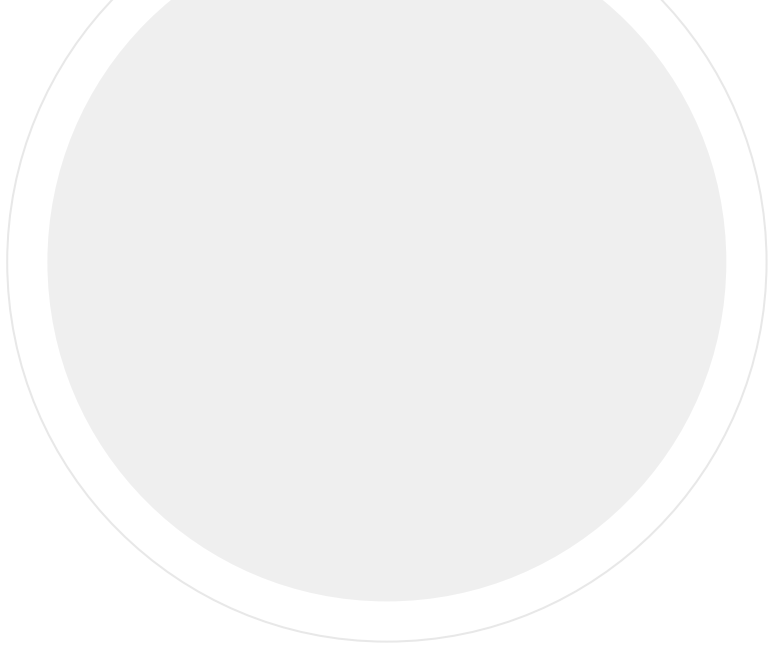
3

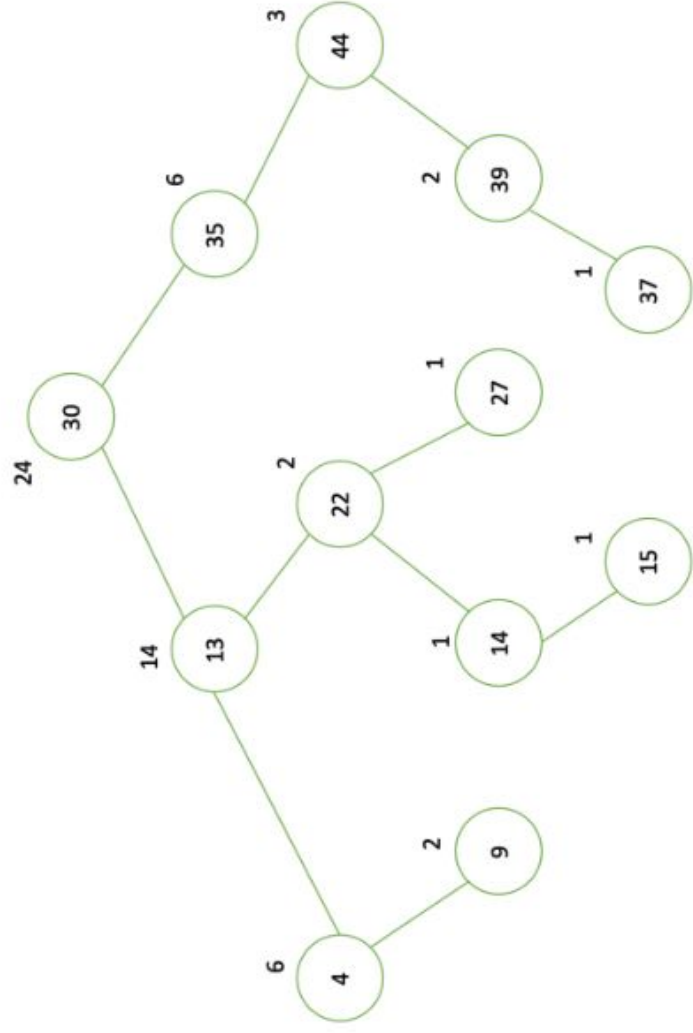
# Une nouvelle structure

Un treap combine les deux approches précédentes. C'est un arbre qui, à chaque noeud, stocke:

- docId, la **clé**
- Fréquence, la **priorité**

C'est un arbre binaire selon la clé, un min-heap selon la priorité

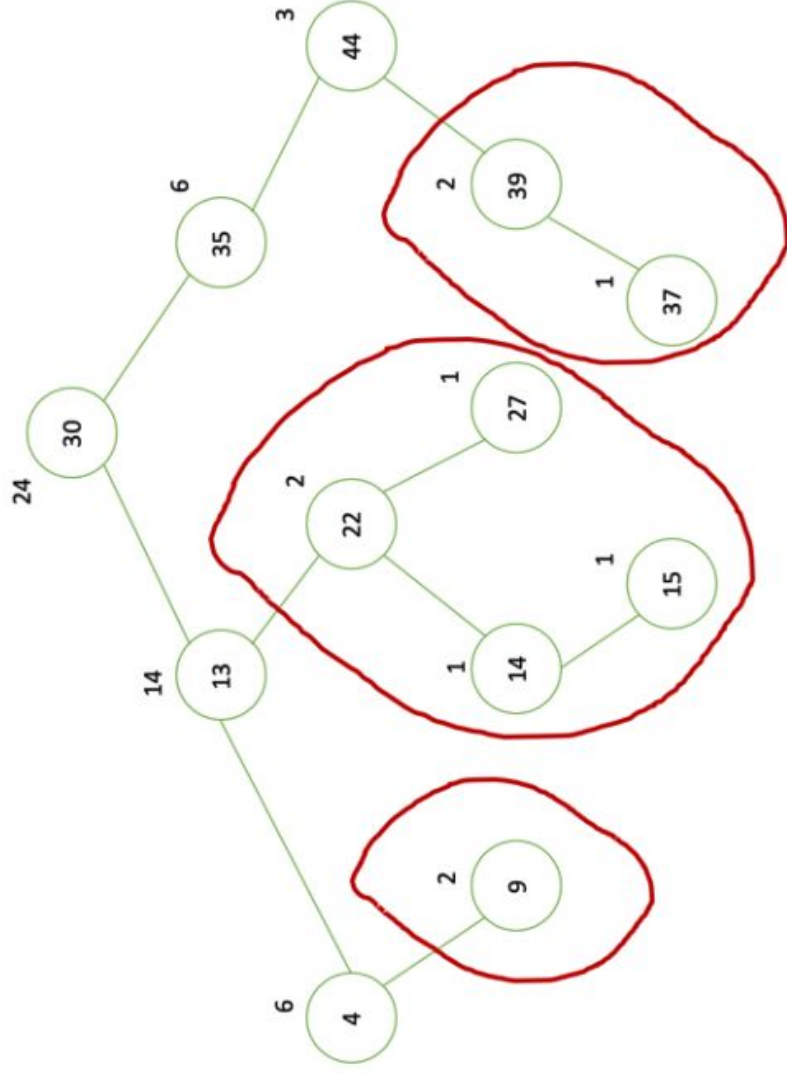




La racine se situe en haut



12



Réduction de l'arbre par loi de Zipf. Les zones rouges ont des fréquences inférieures à une fréquence bien choisie et sont stockées à part.

# Process de requête

4

# Parcours de treaps

## Un treap par terme

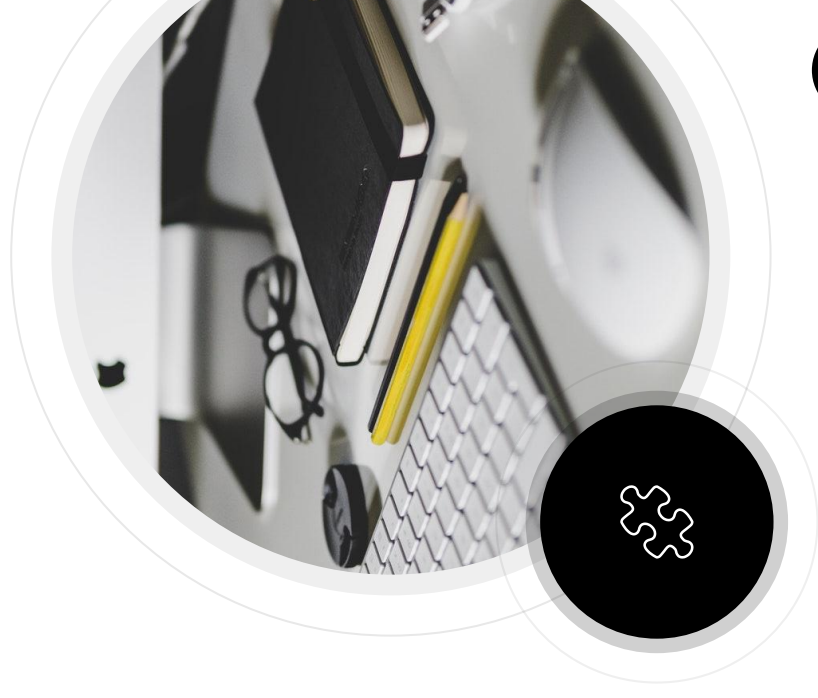
Pour chaque treap, on maintient une pile de noeuds, et un curseur.

## Globalement

On maintient un curseur **vt** et un doc **d** vers lequel on va,

Une borne inférieure de score **L** pour la pile de **k** documents,

Une borne supérieure **U** au score du document **d** grâce calculée avec le curseur **vt**



# Résultats

5



# Gains

Espace disque  
18% par rapport à  
l'approche Block Max  
sur laquelle est basée  
cet article

Temps de requête  
Entre 5 et 15%

Sur une collection de  
25,2 millions de  
documents

Sur des requêtes de -  
de 5 termes,  
demandant les 20  
meilleurs résultats



**Questions?**