

# Arbres de Binaires de Recherches Compressés

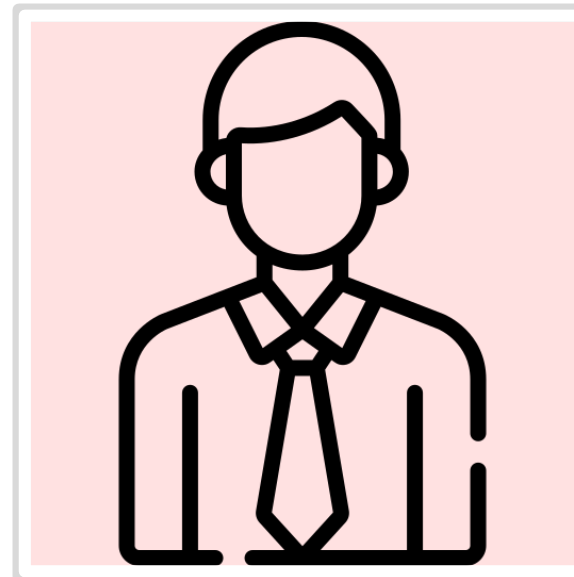


Devoir de programmation Ouverture

# Membres



HADEJBAR Rabah



LOUNIS Lyes



LEFFAD Karim

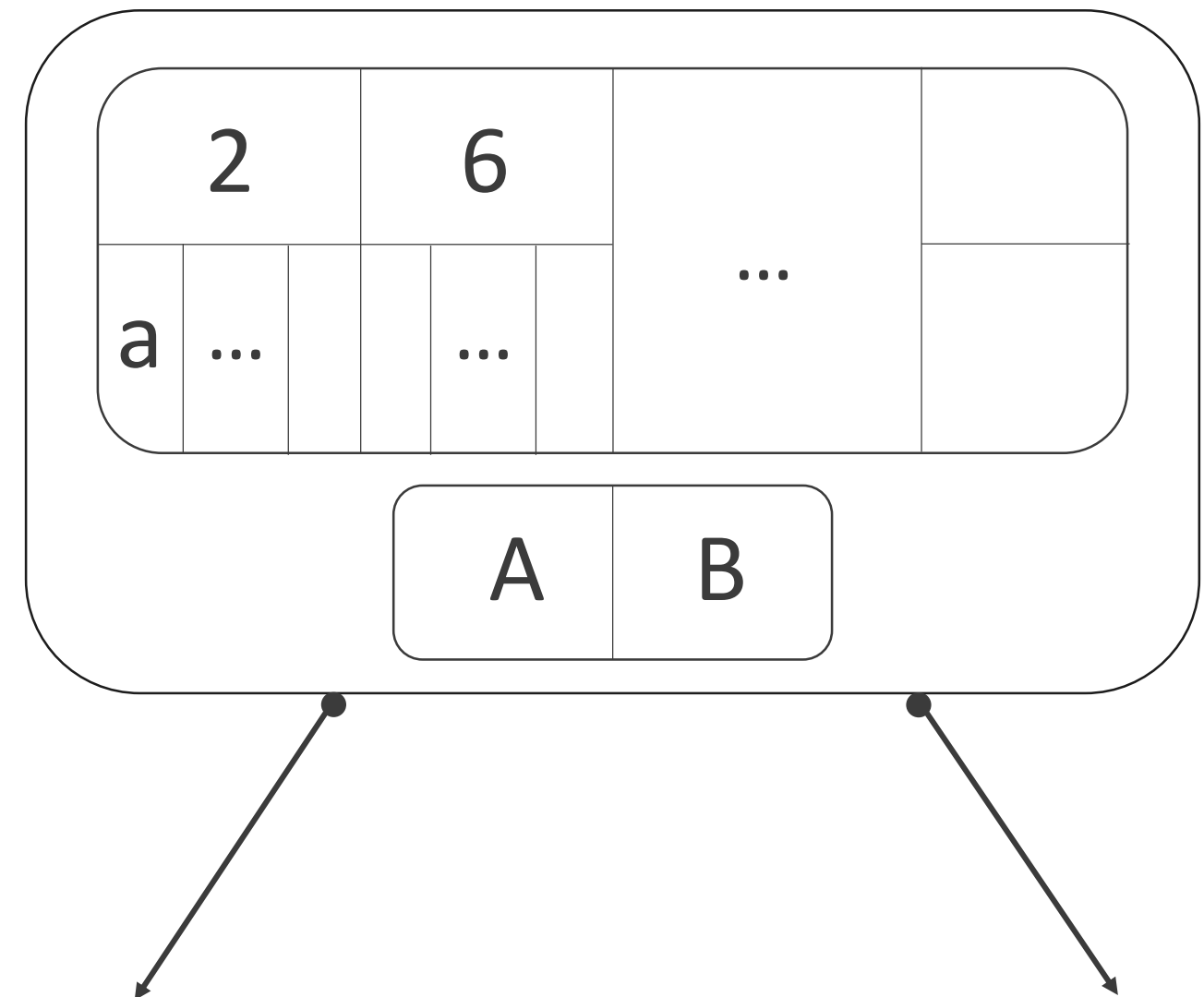
# Notre Structure



# Structure de l'arbre compressé



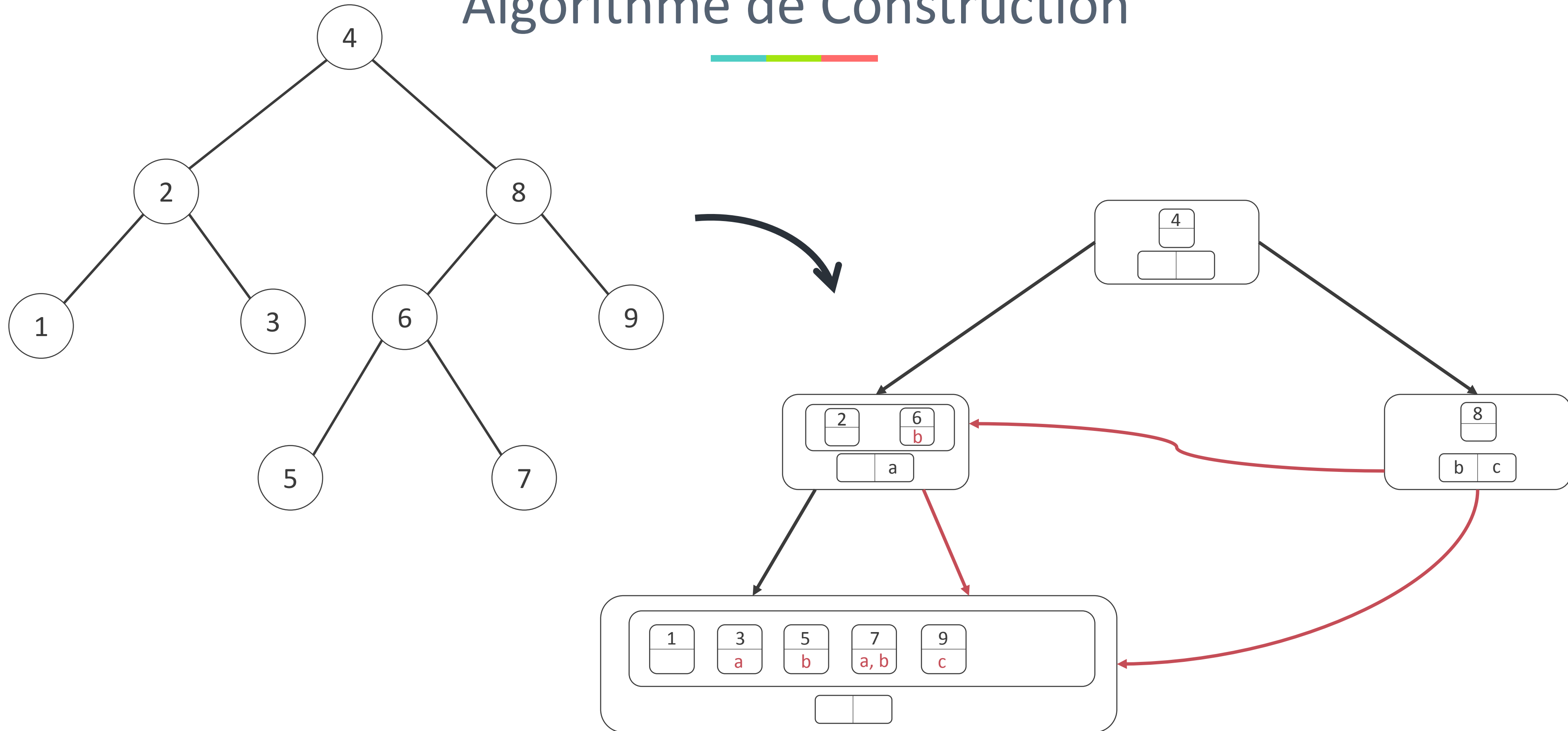
```
type ('a,'b) pointeurAbr = Null | Pointeur of ('a, 'b) abrComprime ref
and ('a,'b) abrComprime = EmptyCL | NodeC of ('a, 'b) contentNode
and ('a,'b) contentNode = {
  mutable valuesNode: 'a;
  labelArcsRouge: ('b * 'b);
  abrGauche: ('a,'b) pointeurAbr ;
  abrDroite: ('a,'b) pointeurAbr
}
```



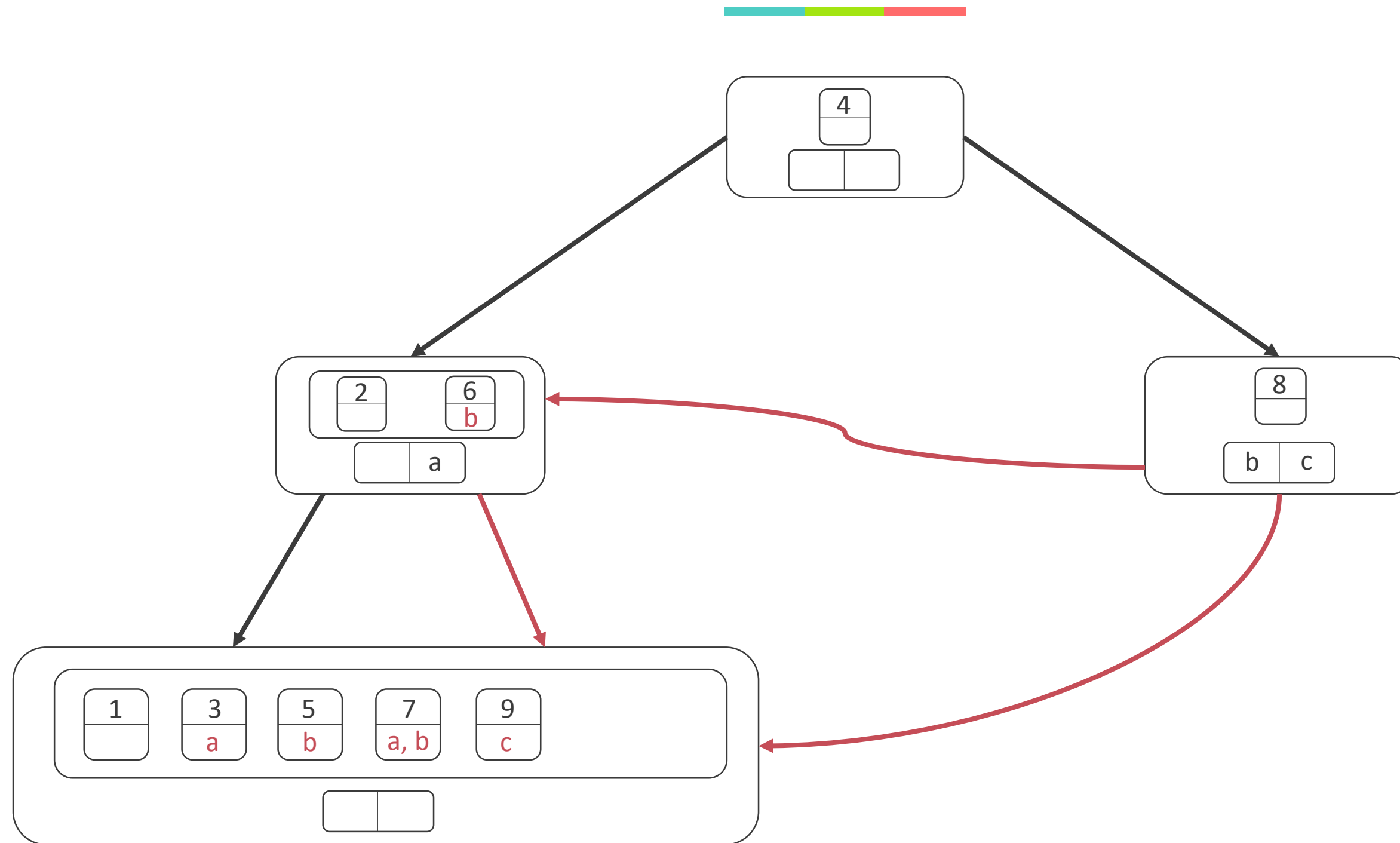
# Nos Algorithmes



# Algorithme de Construction



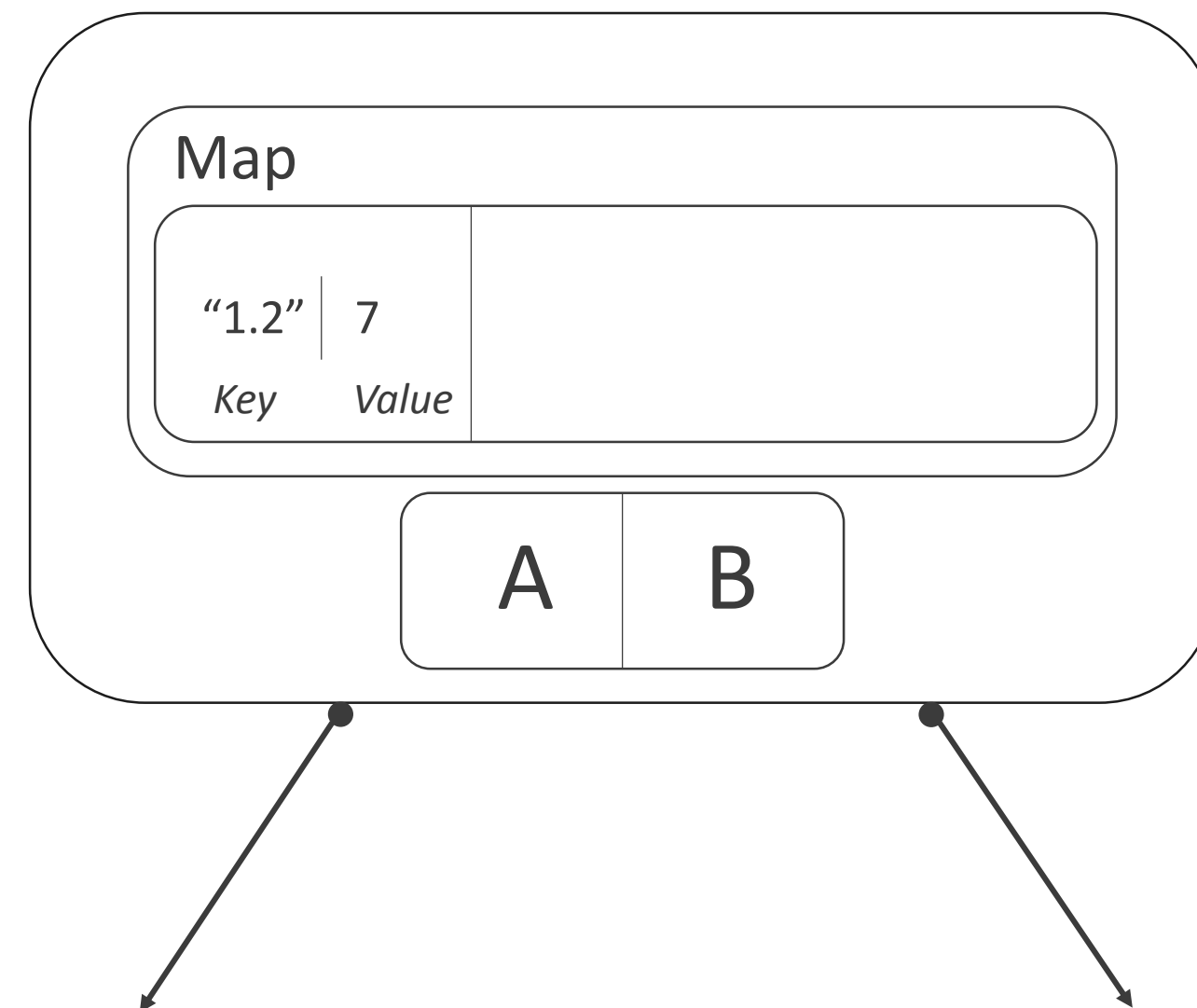
# Algorithme de Recherche



Complexité =  $O(n * \log(n)^2)$

# La compression avec une *Map*

```
module M=Map.Make(String);;
```



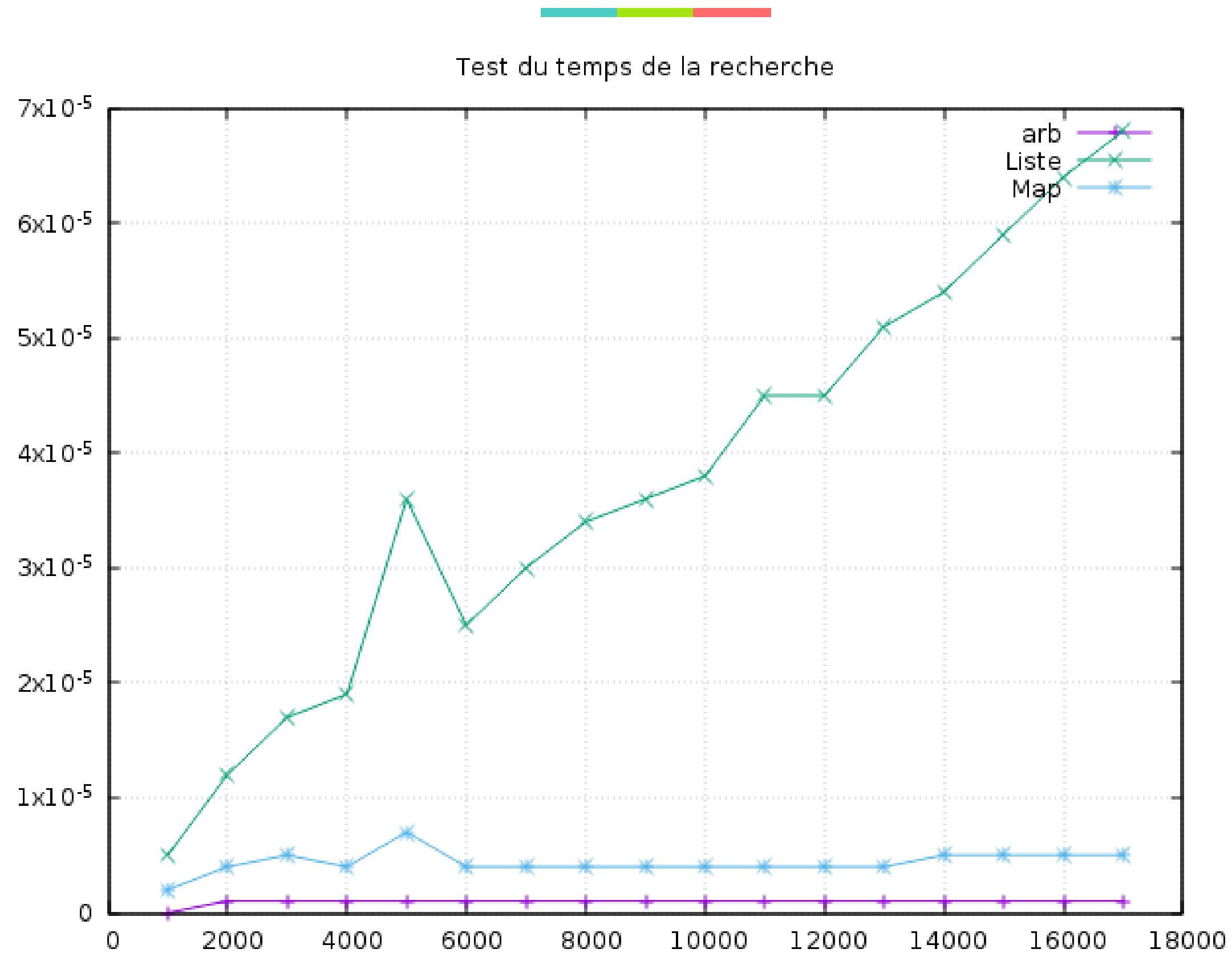
Complexité =  $\ominus$  ( hauteur de l'arbre \*  $\log(n)$  )



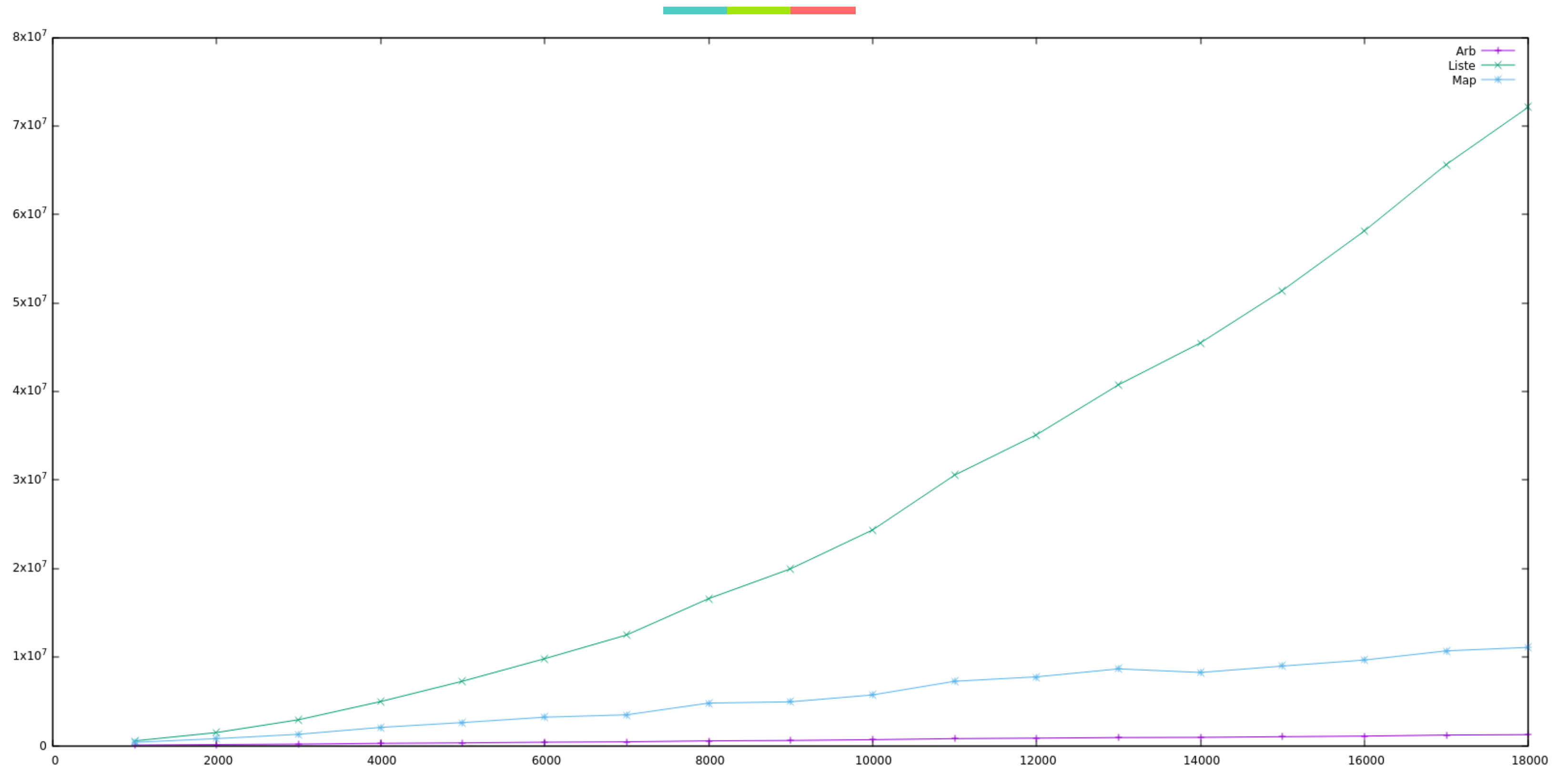
# Expérimentations



# Complexité en temps des algorithmes de recherche



# Complexité en espace du gain de la compression des ABR



# Merci



Des questions ?