

Algorithmique Avancée

Devoir de programmation : Tries

Dat NGUYEN
Tomohiro ISHIWATA
2015-2016

Structures de données

Arbre de la Briandais :

```
public class BRDtree {  
  
    protected Character key;  
    protected BRDtree child;  
    protected BRDtree next;  
}
```

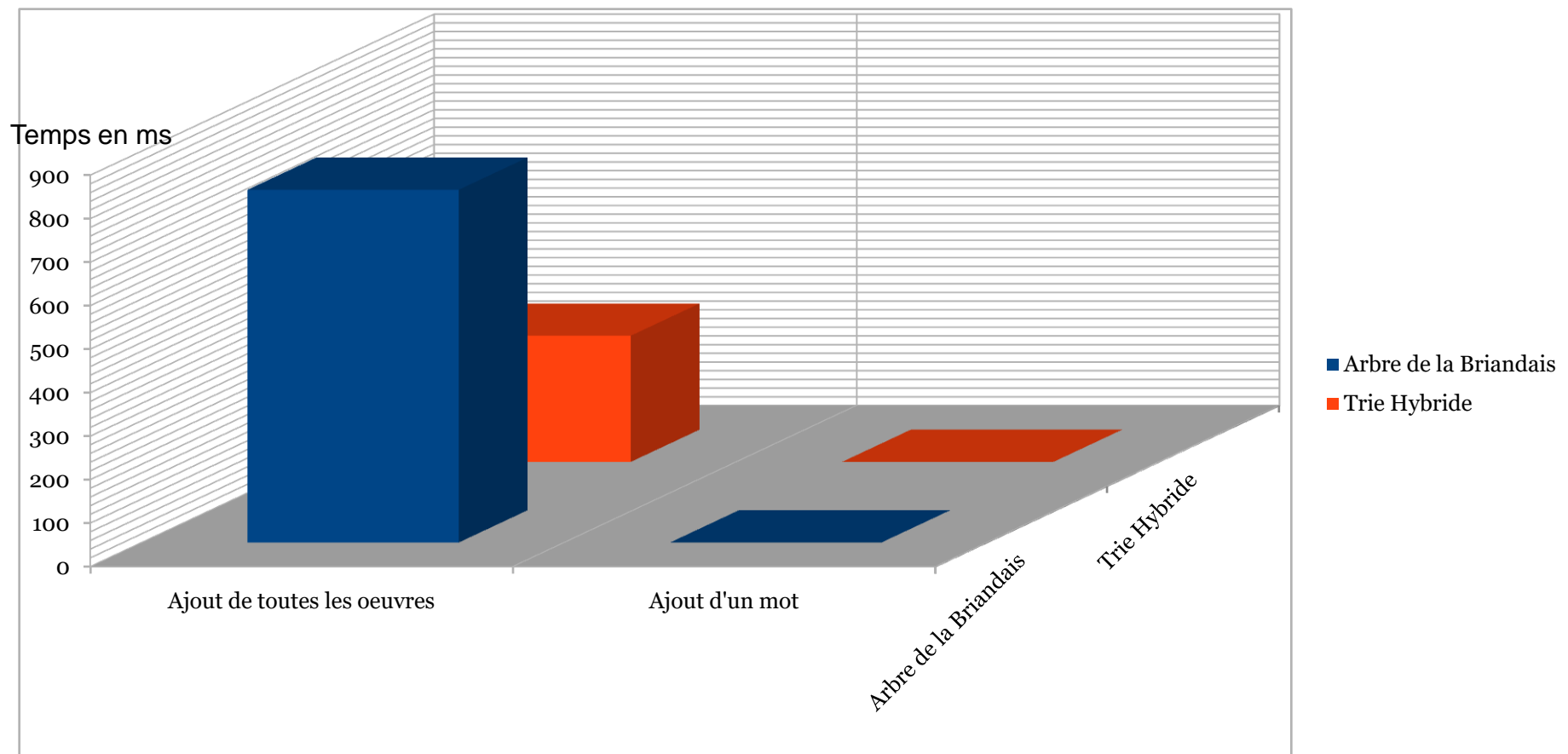
Caractère de fin de mot utilisé est '\0'

Trie Hybride :

```
public class HBDtree {  
  
    protected Character car;  
    protected Character val;  
    protected HBDtree inf;  
    protected HBDtree eq;  
    protected HBDtree sup;  
}
```

Comparaison

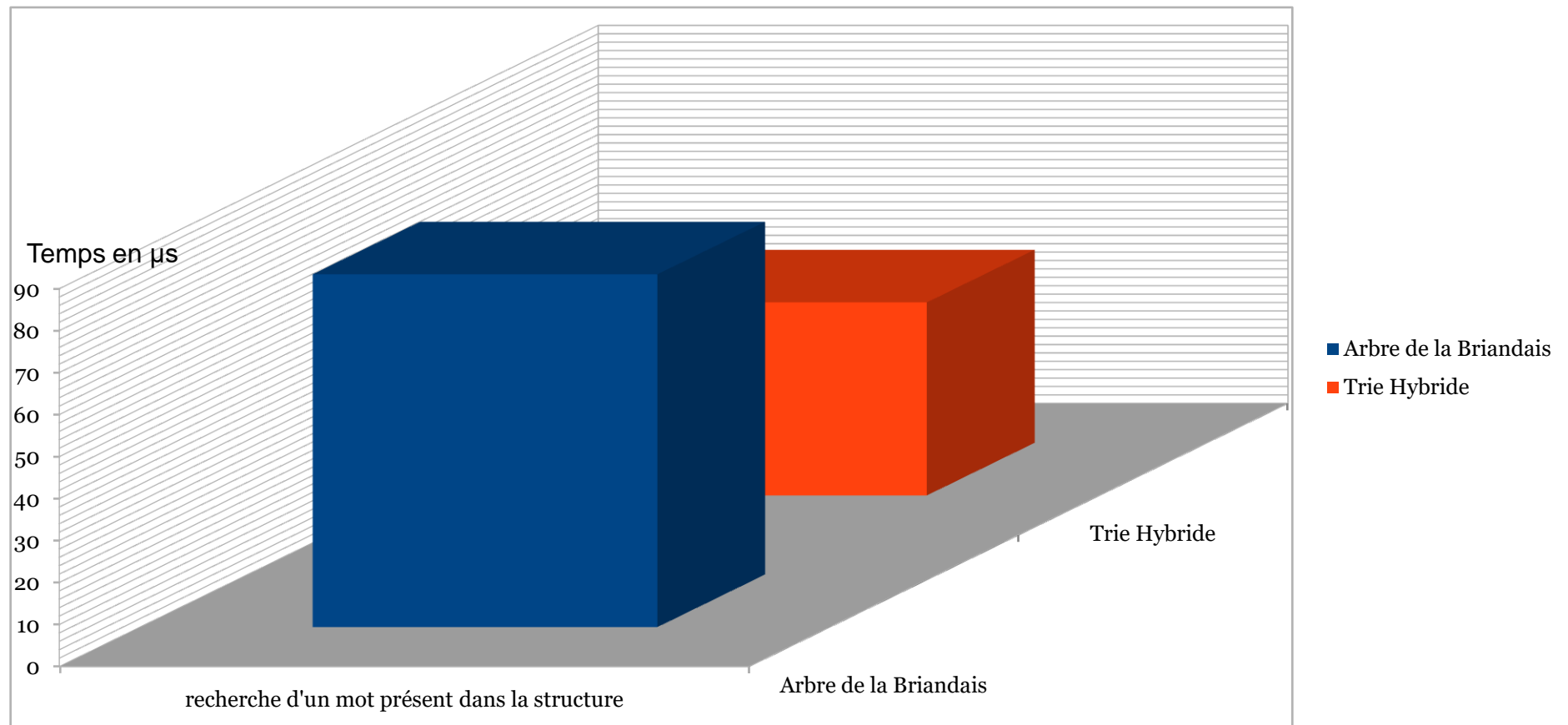
Ajout d'un ensemble d'œuvres et d'un mot



Complexité en $O(\text{longueurMot} * \text{tailleAlphabet})$ dans les deux cas

Comparaison

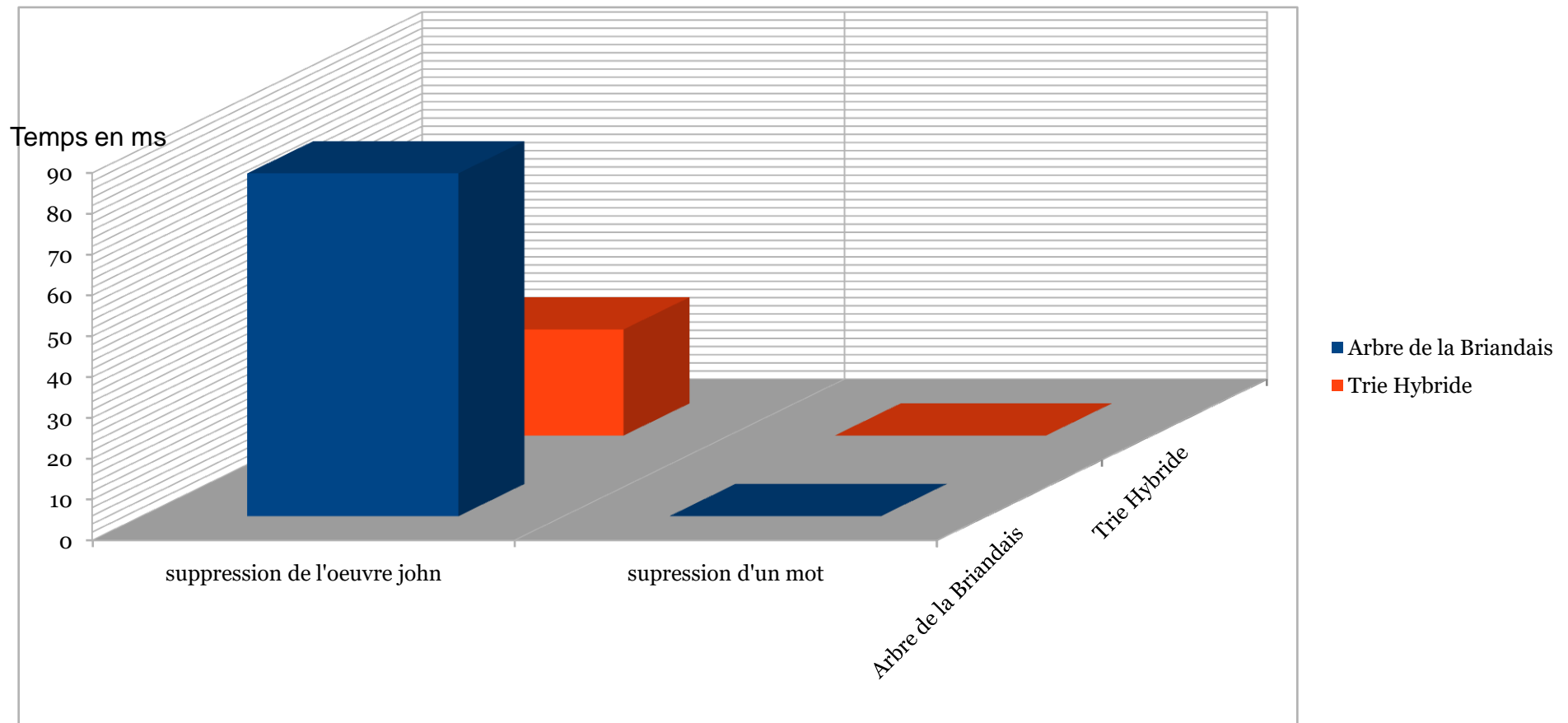
Recherche d'un mot présent dans les deux arbres



Complexité en $O(\text{longueurMot} * \text{tailleAlphabet})$ dans les deux cas

Comparaison

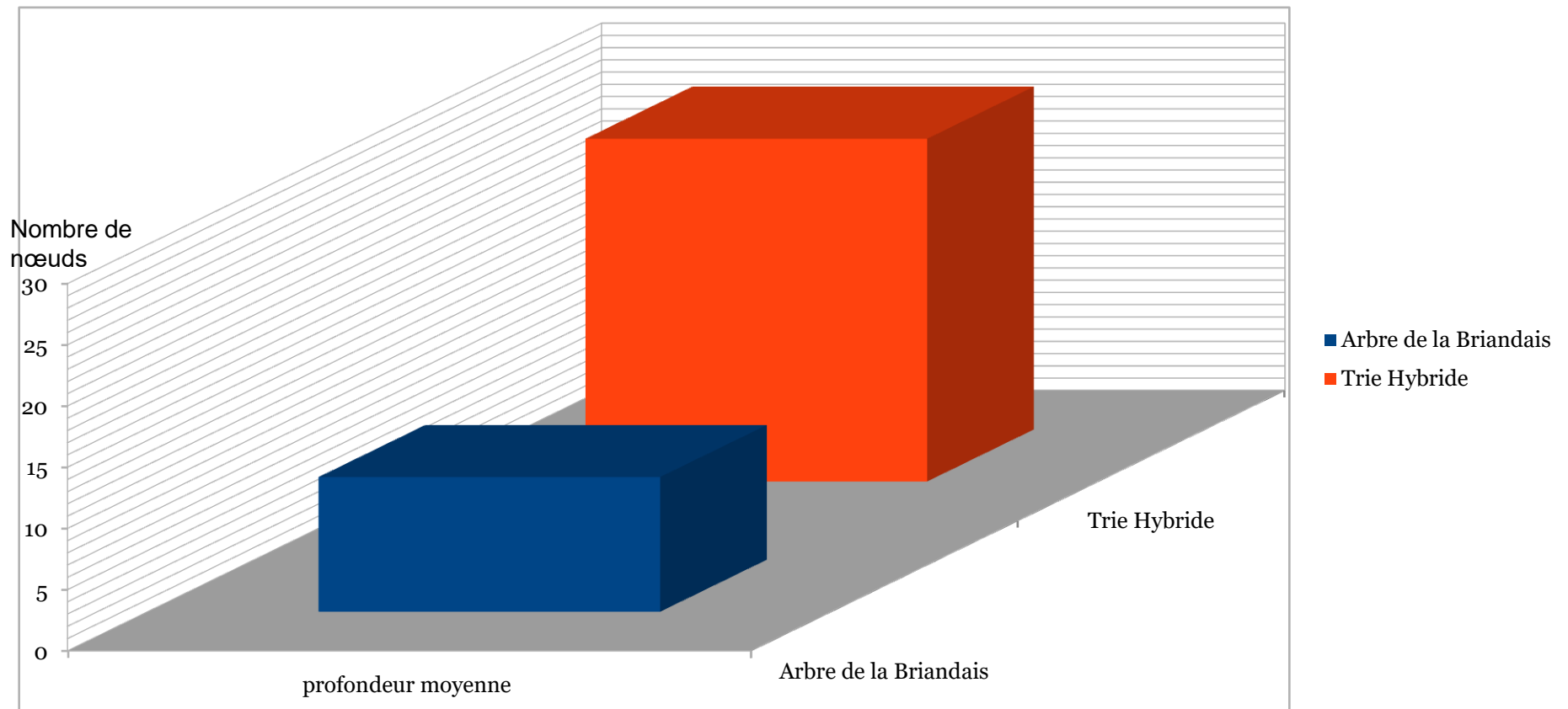
Suppression de tous les mots d'une œuvre et suppression d'un seul mot



Complexité en $O(\text{longueurMot} * \text{tailleAlphabet})$ dans les deux cas

Comparaison

Profondeur moyenne de chacune des structures, après ajout de l'ensemble des œuvres de Shakespeare



Complexité en $O(\text{tailleAlphabet})$ dans les deux cas

Conclusion

- Pour représenter un dictionnaire contenant un nombre conséquent de mots, on optera plutôt pour un trie hybride, car la performance en temps est minimisée pour différentes opérations (recherche, suppression...).
- Qu'en est-il de l'espace occupé par ces différentes structures ?
On peut supposer que l'arbre de la Briandais occupe moins d'espace que le trie hybride :
 - > en comparant leur profondeur moyenne
 - > en comparant leur structure respective
 - > Briandais : 1 caractère et 2 références sur BRDtree
 - > Hybride : 2 caractères et 3 références sur HBDtree