

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche principale	Fonctionnalité #1
Problématique : Il faut avoir un champ de recherche principal performant car les utilisateurs veulent une recherche rapide / presque instantanée.	

Option 1 : Utilisation des méthodes de l'objet array	
Avantages : <ul style="list-style-type: none">- Code plus maintenable : Moins de code, plus modulaire, plus facile à modifier- Code plus lisible : Exprime clairement l'intention du code de manière concise	Inconvénients : <ul style="list-style-type: none">- Moins performant sur de grosses quantités de données- Moins flexible : Difficile d'optimiser précisément le parcours des données
Performance sur jsben.ch : 50 recettes: 883480 → 99.15% 200 recettes : 255079 → 98.21% 500 recettes : 101541 → 97.73% 1000 recettes : 49982 → 98.65%	

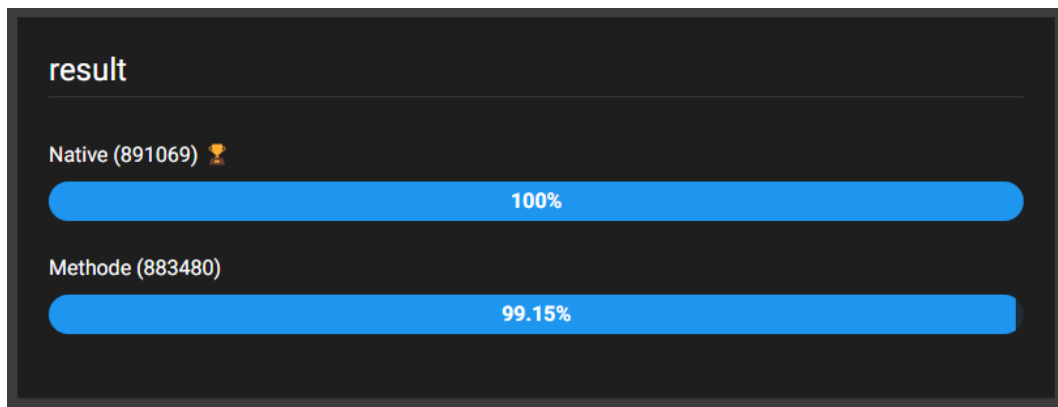
Option 2 : Utilisation des boucles natives	
Avantages : <ul style="list-style-type: none">- Optimisation : Effectuent un seul parcours du tableau, ce qui optimise l'exécution.- Légèrement plus performant sur de grosses quantités de données- Contrôle total : Possibilité d'optimiser (break, continue)	Inconvénients : <ul style="list-style-type: none">- Moins lisible : Code plus long et plus complexe.- Maintenance plus difficile : Modifications plus risquées et lourde
Performance sur jsben.ch : 50 recettes: 891069 → 100% 200 recettes : 259730 → 100% 500 recettes : 103895 → 100% 1000 recettes : 50667 → 100%	

Solution retenue : Comparaison des performances :

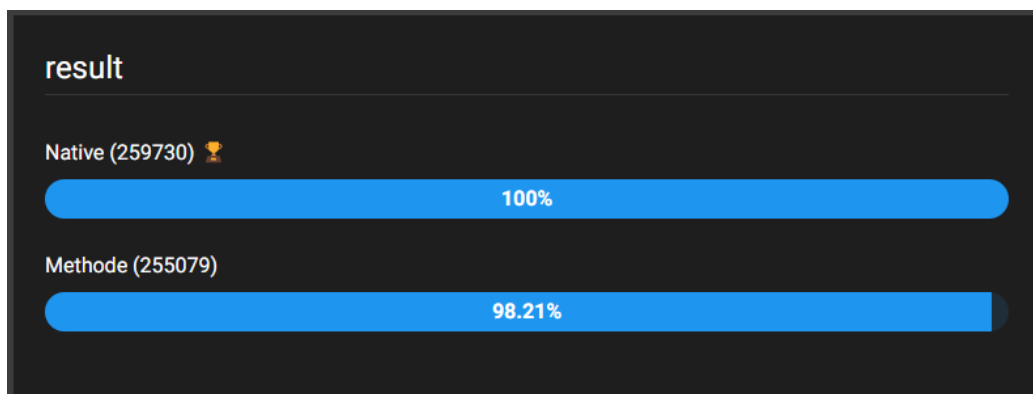
	Native	Méthode
50	891069	883480
200	259730	255079
500	103895	101541
1000	50667	49982

Suite aux tests de performance, la méthode la plus adaptée pour l'implémentation finale est l'option 2 : l'utilisation des boucles natives

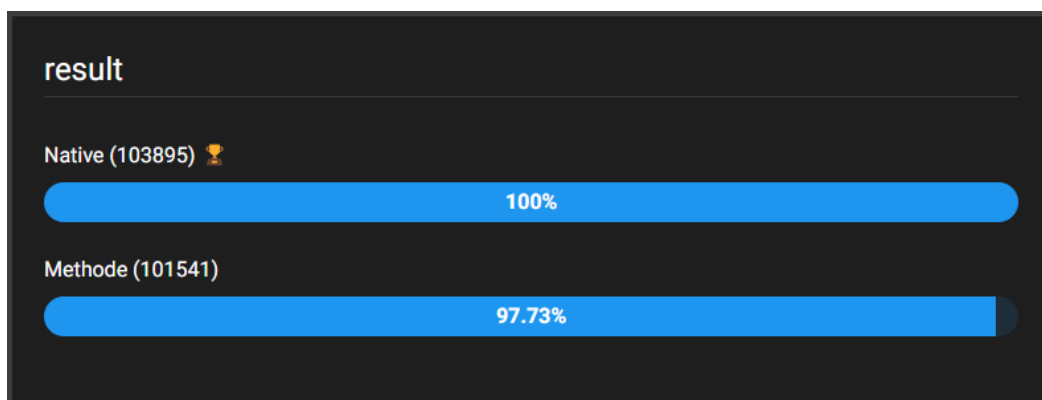
50 :



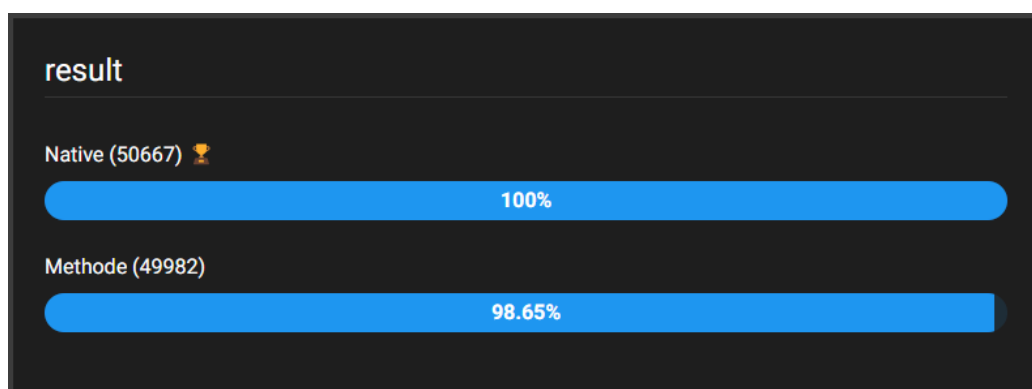
200 :



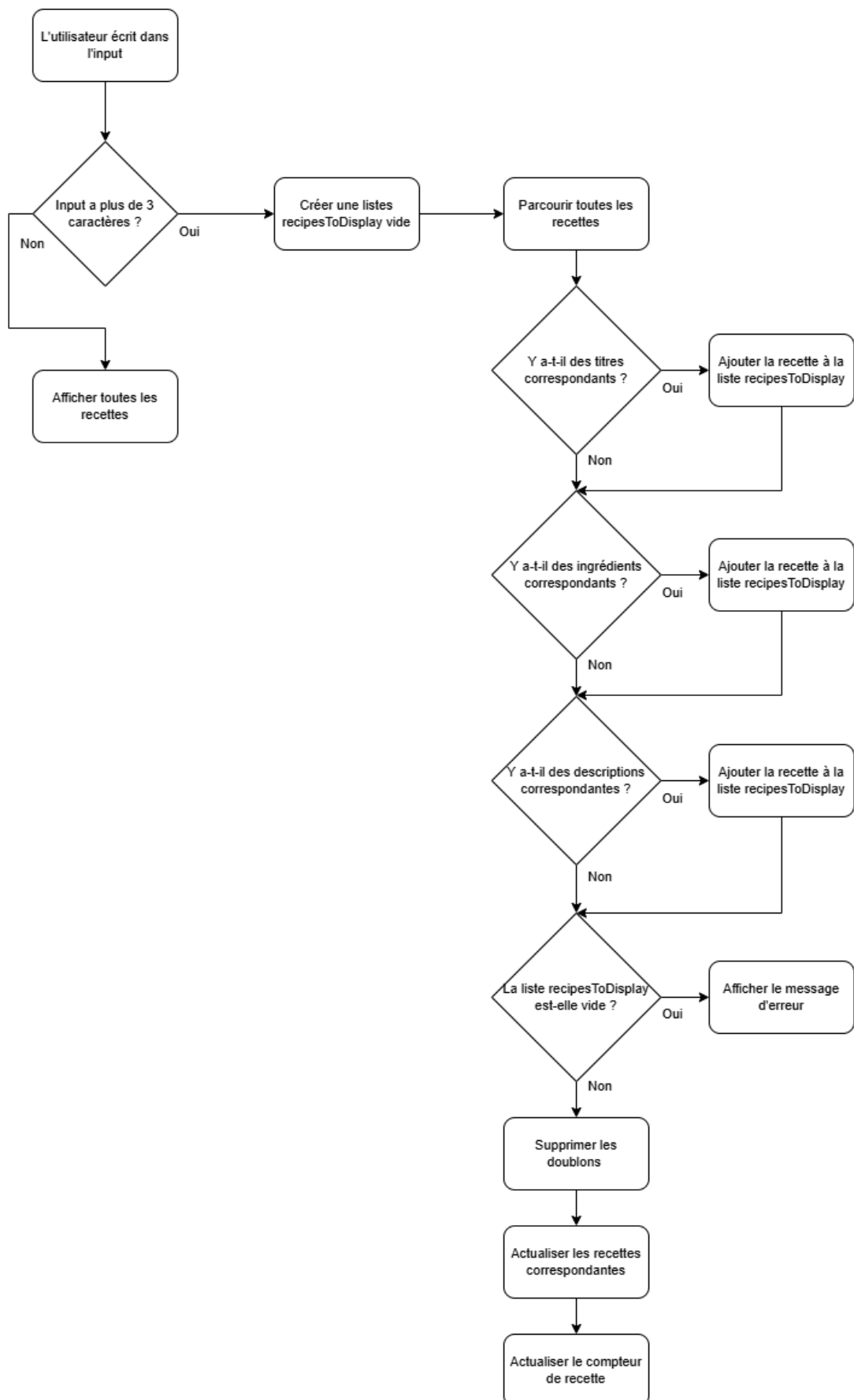
500 :



1000 :



Annexes



<https://jsben.ch/Bbukm>

	Native	Méthode
50	891069	883480
200	259730	255079
500	103895	101541
1000	50667	49982