Fiche d'investigation de fonctionnalité

Fonctionnalité : algorithme de recherche par input	**Fonctionnalité**
Problématique : Afin de pouvoir retenir un maximum d'utilisateurs, nous cherchons à obtenir le plus rapidement possible des résultats cohérents avec l'input fourni.	

Option 1 : Classic for loop

Dans cette option,nous avons un algorithme basé sur l'utilisation d'une boucle forloop. L'avantage de ce système tient en sa facilité d'écriture, ainsi qu'à sa vitesse d'exécution potentiellement un peu plus élevée. Il en résulte néanmoins un code plus verbeux, et donc moins facilement maintenable. Par ailleurs, aucun gain de performance n'a été constaté ici.

Avantages	**Inconvénients**
⊕ Performances pouvant être légèrement supérieures. ⊕ Facile à mettre en place car logique connue de tous.	⊖ Code potentiellement moins maintenable, moins concis, qui souffrira peut-être d'une complexification future.

Vitesse de l'algorithme :

Entre 134825 et 142058 opérations par seconde sur jsben.ch.

Option 2 : Approche fonctionnelle, avec la méthode .foreach() de l'objet Array.

Dans cette option, on extrait du tableau recipes les recettes correspondant à notre input. Ces résultats vont directement peupler un nouveau tableau nommé "searchResults" grâce à push. Le code est plus concis, et sa logique abordable au premier coup d'œil.

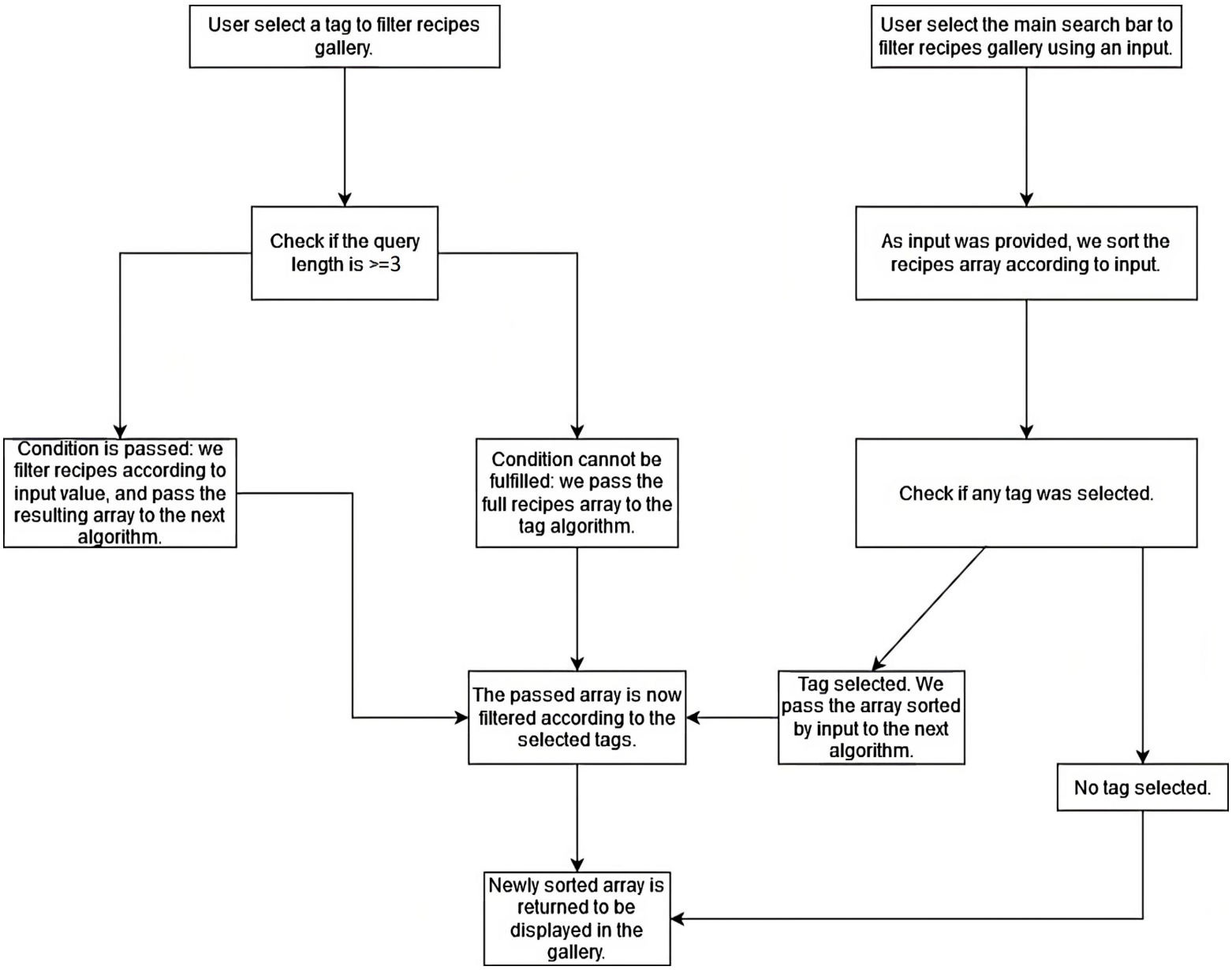
Avantages	**Inconvénients**
⊕ Fonction plus courte. ⊕ Sens du code aisément abordable, ce qui facilite la maintenance future et les mises à jours éventuelles.	⊖ En théorie, les performances peuvent s'avérer inférieures à une boucle for. En l'espèce, ce n'est pas le cas : Les deux versions de l'algorithme délivrent des performances très similaires. ⊖ Moins personnalisable qu'une for loop : on ne peut pas stopper prématurément une itération (avec break), ou itérer à l'envers d'un tableau directement.

Vitesse de l'algorithme:

Entre 133095 et 143182 opérations par seconde sur jsben.ch.

Solution retenue :

Une fois pris en compte le contexte de la mission, et les résultats du benchmark, il me semble approprié de retenir l'approche fonctionnelle, avec forEach. Le gain de performance qu'aurait théoriquement pu apporter forloop ne s'est pas réalisé. Les deux méthodes offrent des résultats très similaires, et la maintenabilité future du code est un point important.



Comparaison entre les deux algorithmes sur jsben.ch, le code block 2 correspondant à la version utilisant les méthodes de l'objet array, et le block 1 aux for loop classiques :