

# Fiche d'investigation de fonctionnalité

# Comparaison d'un algorithme

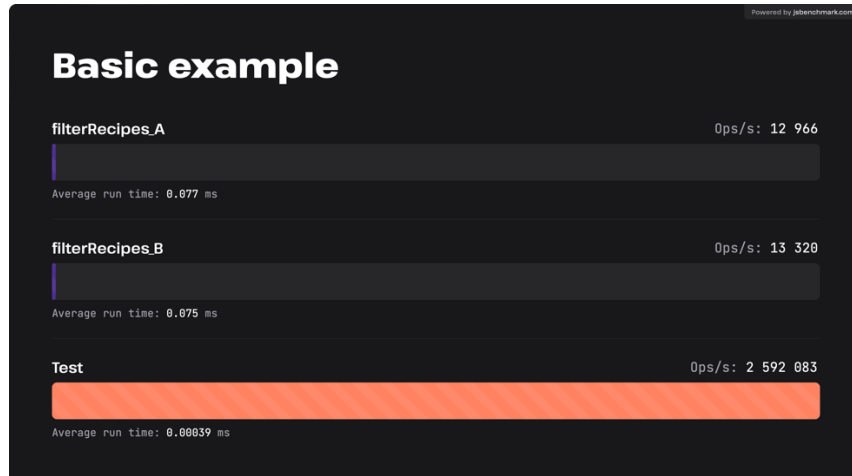
<b>Fonctionnalité :</b> Tri des données du champs de recherche principal	
<b>Problématique :</b> Choisir l'algorithme qui permettra une recherche la plus rapide et la moins nécessiteuse en ressources	

<b>Solution A :</b> Programmation fonctionnelle	Cette option utilise les méthodes de l'objet Array comme foreach, filter, map, reduce...
<b>Avantages :</b> <ul style="list-style-type: none"><li>• Code plus lisible et expressif</li><li>• Favorise l'immutabilité, ce qui réduit les effets de bord.</li><li>• Possibilité d'exécuter certaines opérations en parallèle (optimisation potentielle par le moteur JavaScript).</li><li>• Plus moderne et cohérent avec les paradigmes de programmation actuels</li></ul>	<b>Inconvénients :</b> <ul style="list-style-type: none"><li>• Peut être plus gourmand en ressources (création d'intermédiaires en mémoire).</li><li>• Moins de contrôle sur le flux d'exécution (chaque méthode crée un nouveau tableau).</li><li>• Performances parfois inférieures aux boucles natives pour les grands volumes de données.</li></ul>
<b>Temps d'exécution moyen : 0.077 ms</b>	Donnée recherché : "Coco", ["lait_de_coco", "jus_de_citron"], ["blender"], ["verres"]

<b>Solution B :</b> Boucles natives	Cette option propose d'utiliser les boucles natives comme while, for...
<b>Avantages :</b> <ul style="list-style-type: none"><li>• Exécution plus rapide dans la plupart des cas, surtout sur de grandes quantités de données.</li><li>• Meilleur contrôle sur le flux d'exécution et la gestion de la mémoire.</li><li>• Pas d'allocations inutiles (évite la création de tableaux intermédiaires).</li></ul>	<b>Inconvénients :</b> <ul style="list-style-type: none"><li>• Code plus verbeux et parfois plus difficile à maintenir.</li><li>• Risque plus élevé d'effets de bord si la gestion de l'état n'est pas rigoureuse.</li><li>• Moins intuitif que les méthodes fonctionnelles pour certaines manipulations complexes</li></ul>
<b>Temps d'exécution moyen : 0.075 ms</b>	Donnée recherché : "Coco", ["lait_de_coco", "jus_de_citron"], ["blender"], ["verres"]

<b>Solution retenue :</b> <p>La fiche indique que la solution B a été choisie pour sa meilleure performance, bien que la différence avec la solution A soit minime (0.002 ms). Le choix de la solution B est cohérent pour une optimisation pure des performances, mais la faible différence mesurée laisse entendre que d'autres critères (lisibilité, maintenabilité) pourraient être pris en compte.</p>
---

## Annexes : Résultats



## Annexes : Diagramme de l'algorithme

