Fiche d’investigation de fonctionnalité

Fonctionnalités : Recherche	**Fonctionnalité #1**
Problématique : Effectuer une fonction recherche de la manière la plus efficace possible	

Option 1 : Utilisation de boucle for native
Dans cette option, nous n’utilisons que des boucles for natives, légèrement plus rapide que les boucles des méthodes array, qui permettent d’optimiser en manipulant les indices grâce aux commandes break et continue

Avantages :	**Inconvénients :**
- Légèrement plus rapide que les boucles méthode array - Optimisation possible grâce aux commande break et continue - Compatibilité avec toutes les versions de JS, même très anciennes	- Code plus long, et moins explicite - Maintenance plus laborieuse pour les débutants - Plus propice aux erreurs

Input pour lancer une recherche : Minimum 3 caractères

Option 2 : Utilisation de boucle des méthodes array (forEach, filter)
Dans cette option, nous n’utilisons que des boucles des méthodes array (forEach, filter), afin d’avoir un code plus clair, plus condensé, et plus explicite.

Avantages :	**Inconvénients :**
- Code moins long, plus explicite - Plus simple à maintenir et corriger - Moins propice aux erreurs	- Moins rapide que les boucles natives - Pas d’optimisation possible - Pas compatibles avec les premières versions de JS

Input pour lancer une recherche : Minimum 3 caractères

Solution retenue :
Même si ça ne se joue à pas grand-chose, c’est l’option 1, utilisant les boucles natives, qui nous donnent les meilleurs résultats. La priorité étant d’avoir la meilleure optimisation possible, c’est cette option là que nous choisirons pour la suite, au détriment de la lisibilité du code

Figure 1 – Utilisation boucle natives

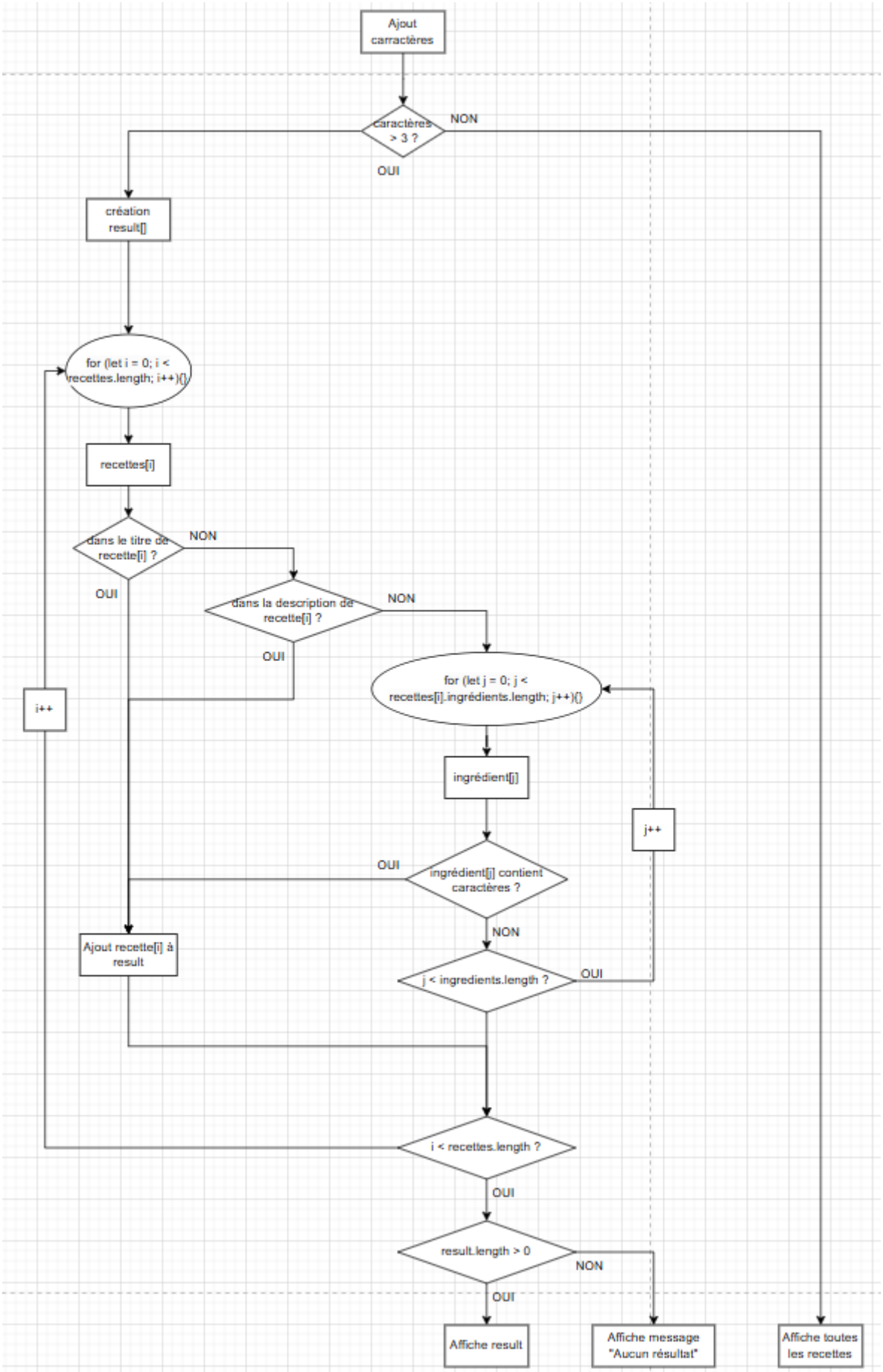
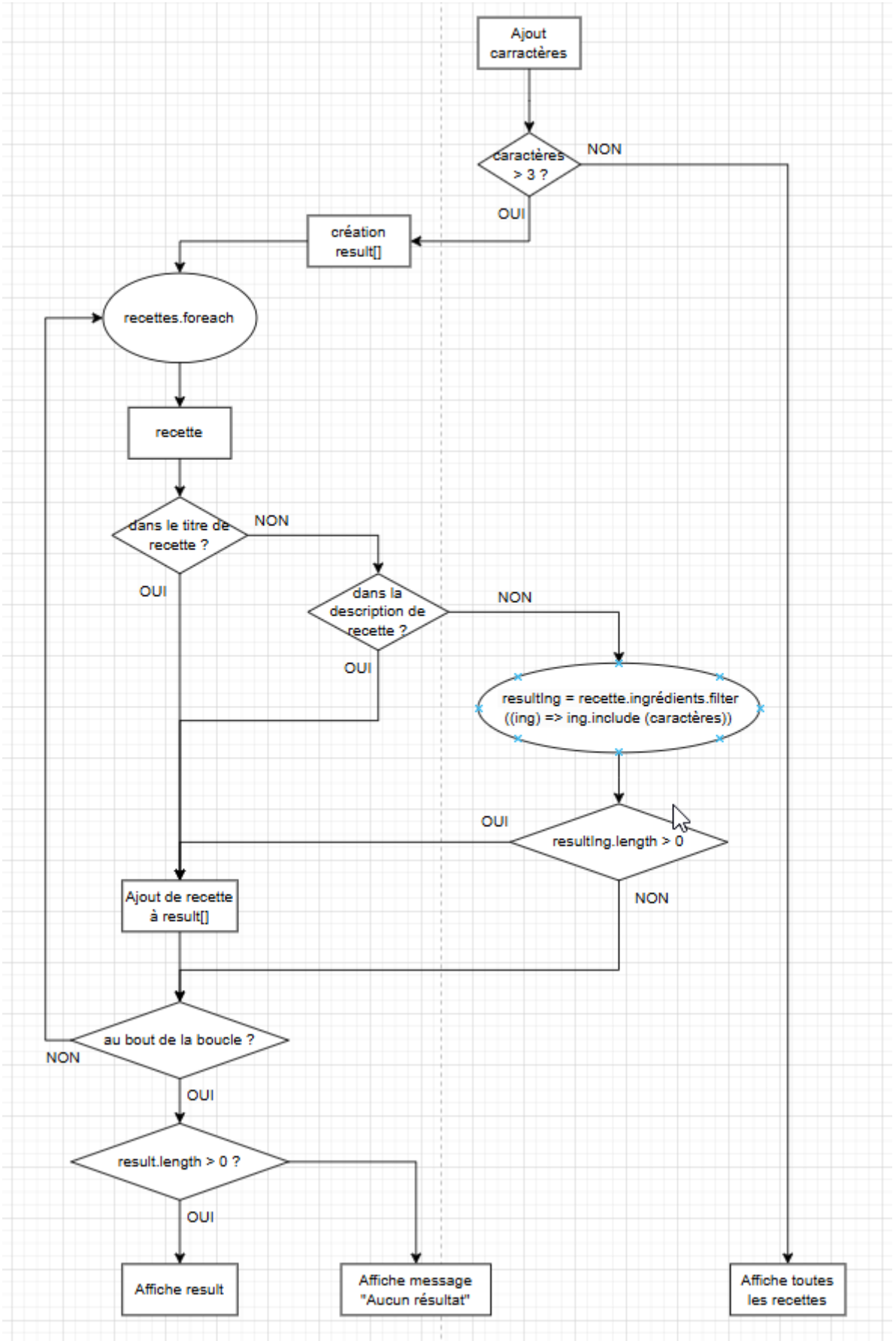


Figure 2 – Utilisation boucle array



Résultats JSBENCH