

Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche recette	Fonctionnalité #1
Problématique : Afin d'éviter un taux de rebond important, la recherche doit être la plus rapide/performante possible.	

Option 1 : Boucle native (for) et regex	
Avantages <ul style="list-style-type: none">⊕ Contrôle direct sur l'itération.⊕ Familière en programmation.⊕ Plus facile à déboguer.	Inconvénients <ul style="list-style-type: none">⊖ Plus de ligne de code.⊖ La totalité du tableau de recette est parcourue.

Option 2 : Méthodes de l'objet array (filter, includes)	
Avantages <ul style="list-style-type: none">⊕ Peu de ligne de code.⊕ Facile à lire.⊕ Facile à mettre en place.	Inconvénients <ul style="list-style-type: none">⊖ La totalité du tableau de recette est parcourue.

Solution retenue: La solution retenue est l'option 1: Boucle native (for) et regex

Performance Comparison: Native Loops vs Array Methods for Recipe Search

RUN TESTS

GENERATE PAGE URL

NEW BENCHMARK

Description

This benchmark compares the performance of two versions of a searchRecipes function. The first version uses native for loops and regular expressions to search through recipe names, descriptions, and ingredients. The second version uses array methods like filter, find, and includes for the same task. Both functions are tested using a sample recipe dataset, and the search term must be at least 3 characters long to return results. This comparison will help assess which approach performs better in terms of execution speed and efficiency for recipe filtering.

result

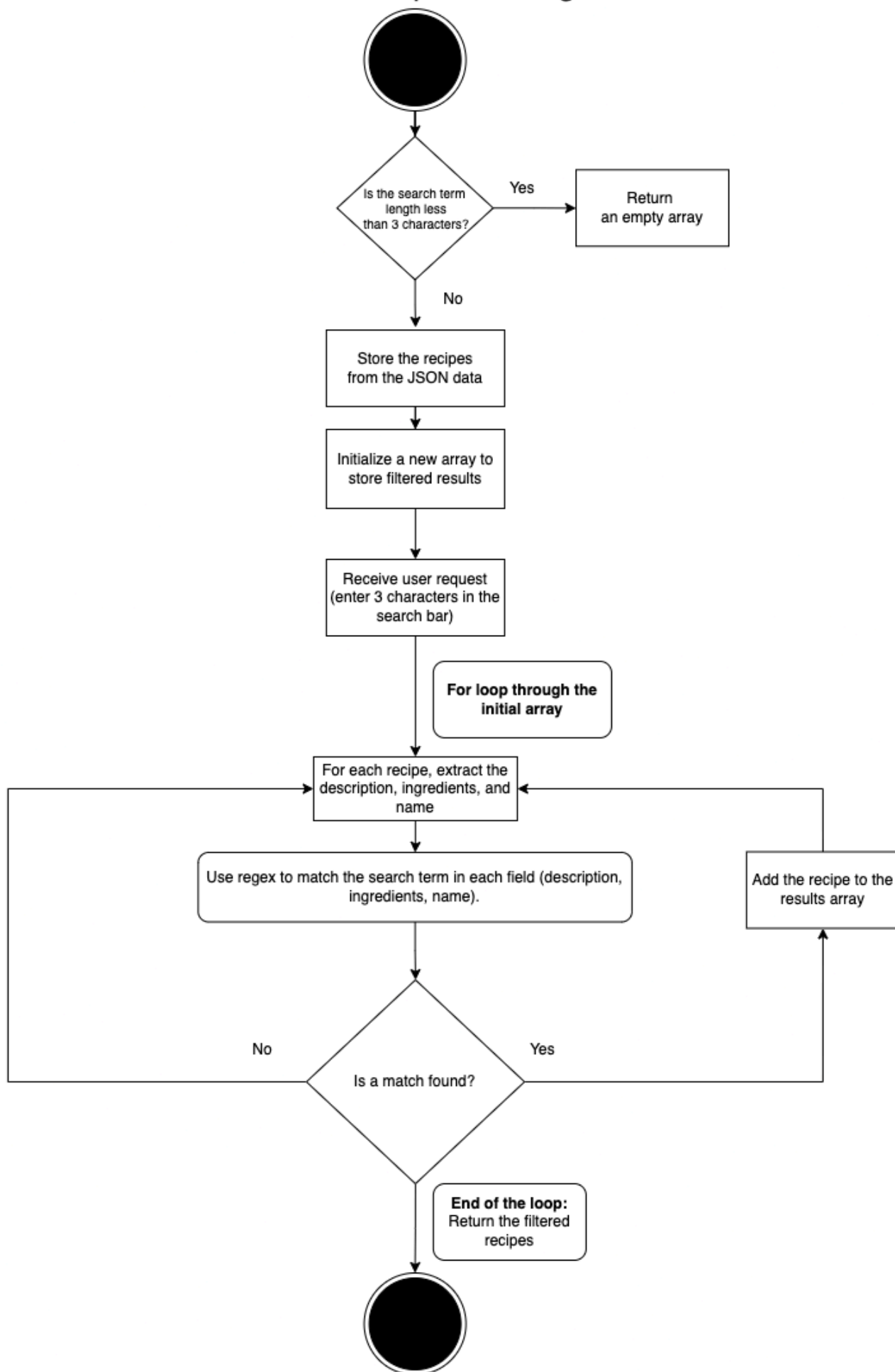
Version 1: Using Native for Loops with Regex (244888) 🏆

100%

Version 2: Using Array Functions (find, filter, includes): (198221)

80.94%

Diagram 1 - SearchRecipes Method Using Native Loops and Regex



**Diagram 2 - SearchRecipes Method Using
Array Functions (filter, find, includes)**

