

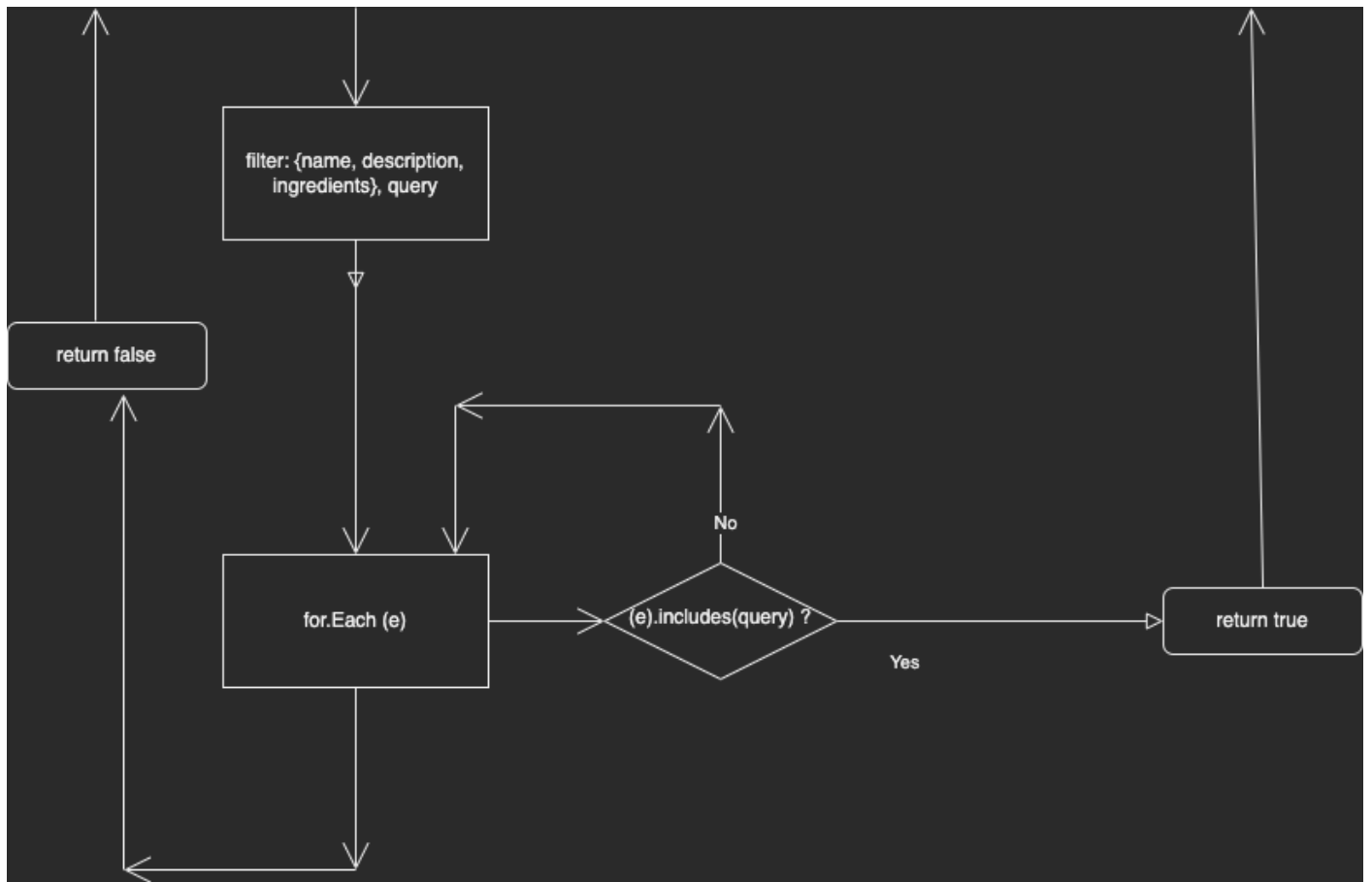
## Fiche d'investigation de fonctionnalité

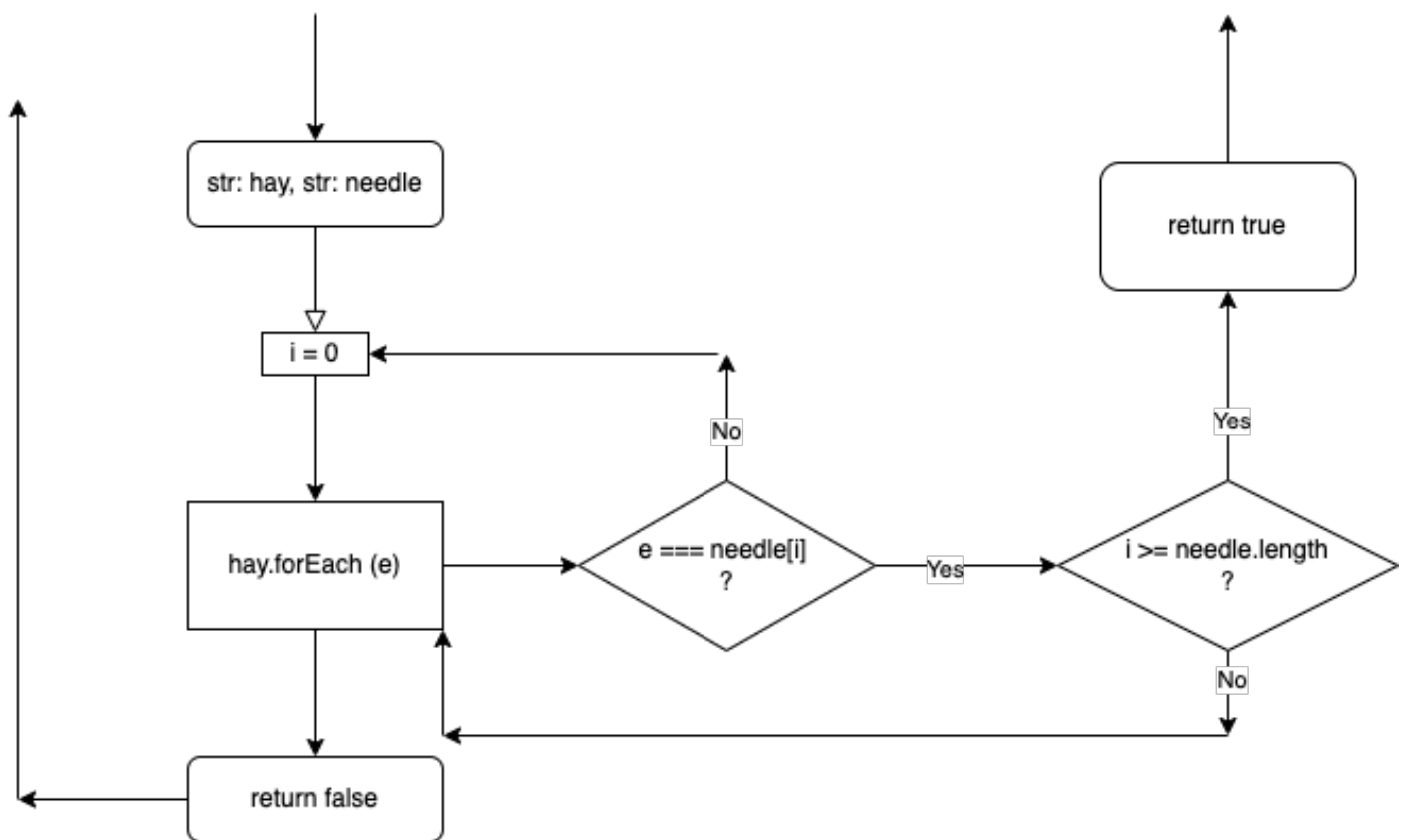
<b>Fonctionnalité</b> : Recherche principale	<b>Fonctionnalité #1</b>
<b>Problématique</b> : Déterminer l'algorithme le plus efficace pour vérifier la présence d'une sous-chaîne de caractères, parmi 3 chaînes aléatoires de type et longueurs variables.	

<b>Option 1 : Méthodes de l'Object Array (Built-in Library)</b> La bibliothèque Js standard inclut de nombreuses méthodes différentes pour l'objet Array. Sur Node & Chrome, l'implémentation du moteur V8 permet de nombreuses optimisations bas niveaux améliorant les performances des constructions haut niveaux à travers différents composants du JIT (Just in Time) compiler.	
<b>Avantages</b> ⊕ Lisibilité humaine, maintenabilité ... ⊕ Rapidité	<b>Inconvénients</b> ⊖ Difficulté d'optimization en fonction du dataSet ? ⊖ Algorithme naïf de complexité $O(m*n)$
<b>Longueur minimal de P: 3</b> Recherche exécuté sur 3 chaînes (T) de taille variable	

<b>Option 2 : Boucles natives</b> Implémentation d'un algorithme sensiblement identique à la méthode includes mais non optimisé par V8 JS étant un langage de haut niveau, l'implémentation d'une construction bas niveau à son échelle du stack sera pour la majorité du temps, dépassé en performance par des méthodes standards	
<b>Avantages</b> ⊕ Personnalisation de l'algorithme de recherche strStr ⊕ Pré Optimisation par adaptation du dataSet ? ⊕ ⊕	<b>Inconvénients</b> ⊖ Moins lisible pour un humain
<b>Même variables d'expérimentation que pour l'option 1</b>	

<b>Solution retenue :</b> Le choix s'est porté sur l'option 1 après analyse des benchmarks Lighthouse sur un env Chrome. L'implémentation de différents algorithmes plus bas niveaux comme KMP ou Boyers Moores peut se présenter comme voie d'amélioration possible.
---





```
[nodemon] starting `node main.js`  
f0:Control loop performance: 241.186ms  
f1:Native loop performance: 210.399ms  
f2:Func loop performance: 260.313ms  
f3:kmp loop performance: 176.107ms  
f4:bm loop performance: 167.793ms
```

```
[nodemon] clean exit - waiting for changes before restart  
[nodemon] restarting due to changes...  
[nodemon] starting `node main.js`
```

```
f0:Control loop performance: 51.606ms  
f1:Native loop performance: 26.521ms  
f2:Functionnal loop performance: 36.46ms  
f3:kmp loop performance: 31.517ms  
f4:bm loop performance: 29.155ms
```

Rechercher  
Bibliothèque  
Créer une playlist  
Titres liés  
Voir épisodes

Recommandations de la semaine  
John Handberg  
Both Sides of the Sky  
Live at Woodstock  
Are You Experienced  
Band Of Gypsies (50t...  
Electric Ladyland  
Axl: Bold As Love  
The Power and The Bla...  
Silicon Valley End Credi...  
Vos titres préférés de 2...  
This Is Prince  
Autour d'Elton

