

## Fiche d'investigation de fonctionnalité

**Fonctionnalités** : recherche linéaire /  
recherche binaire

**Fonctionnalités #1**

**problématique**: dans un fichier JSON/BD, très **vite** trouver concordance entrée par l'utilisateur. Avoir un algorithme efficace.

### Option 1 recherche linéaire

Recherche linéaire (boucle for, while) bien pour BD / JSON de taille abordable.

#### Avantages

- Très simple à mettre en place et à maintenir en cas de refonte BD/JSON.

#### Inconvénient

- pas optimisé atteint vite ses limites pour BD/JSON de grande taille .

Recherche d'un mot ou liste de mots dans une phrase, une liste d'ingrédients ou un texte descriptif sans qu'il n'y ait de difficultés.

### Option 2 recherche binaire

Plus complexe à mettre en place et à maintenir, très efficace sur BD/JSON de très grande taille. Utilise des fonctions récursives.

#### avantages

- Très rapide avec des données de grande taille
- N'analyse pas toute la base de données

#### inconvénient

- Difficile à mettre en place sur des données de type String qui comporte plus d'un mot.
- Nécessite un tableau ordonné sinon.

Sélectionne notre tableau À partir du milieu et vérifie si notre donnée se trouve à droite/gauche du tableau puis réitère l'opération jusqu'à concordance.

## Conclusion

Pour une meilleure performance du site nous recommandons l'utilisation de l'**option 2** (recherche binaire) qui effectue jusqu'à 1 542 986 opération/seconde contre 1 511 105 opération/seconde pour l'option 1 (recherche linéaire).

## Annexes:

figure lier à l'option 1

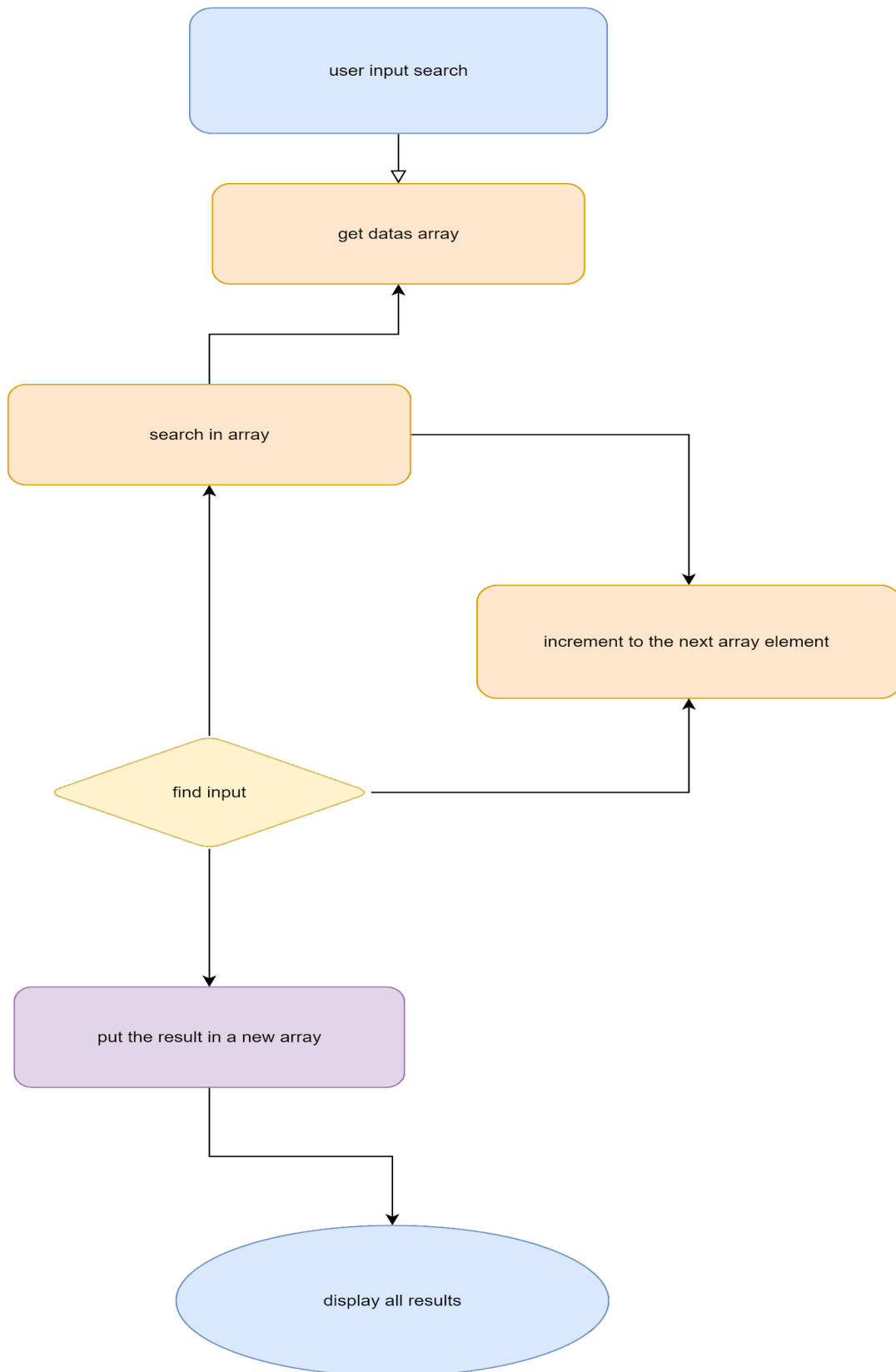


Figure lier à l'option 2

