

Fonctionnalité: Tri/filtre	Fonctionnalité #1
Problématique: Créer une fonctionnalité de recherche la plus fluide et rapide possible dès les premières lettres entrées dans la barre de recherche. Pour cela il faut choisir entre deux algorithmes.	

Option 1: Algorithme tri-fusion (merge sort). Il se base sur le principe de “diviser pour régner”. Sa complexité étant linéarithmique ($O(n \log n)$), il est l’un des plus optimisés pour les traitement de grande séquences de données	
Avantages: <ul style="list-style-type: none"> - Traitement optimal des données - Stable 	Inconvénients: <ul style="list-style-type: none"> - Complexe - Son efficacité dépend du nombre de données (triage egal avant fusion)
Nombres de caractères à saisir : 3 minimum	

Option 2: Algorithme tri-rapide(quick sort). Il se base également sur le principe de “diviser pour régner”. Sa complexité est linéarithmique mais dans le pire des cas, quadratique ($O(n^2)$)	
Avantages: <ul style="list-style-type: none"> - rapide - Le pire des cas est peu probable si correctement mis en oeuvre 	Inconvénients: <ul style="list-style-type: none"> - Instable dans le pire des cas - Moins efficace sur des données déjà partiellement triées (cas particulier)
Nombres de caractères à saisir : 3 minimum	

ANNEXES

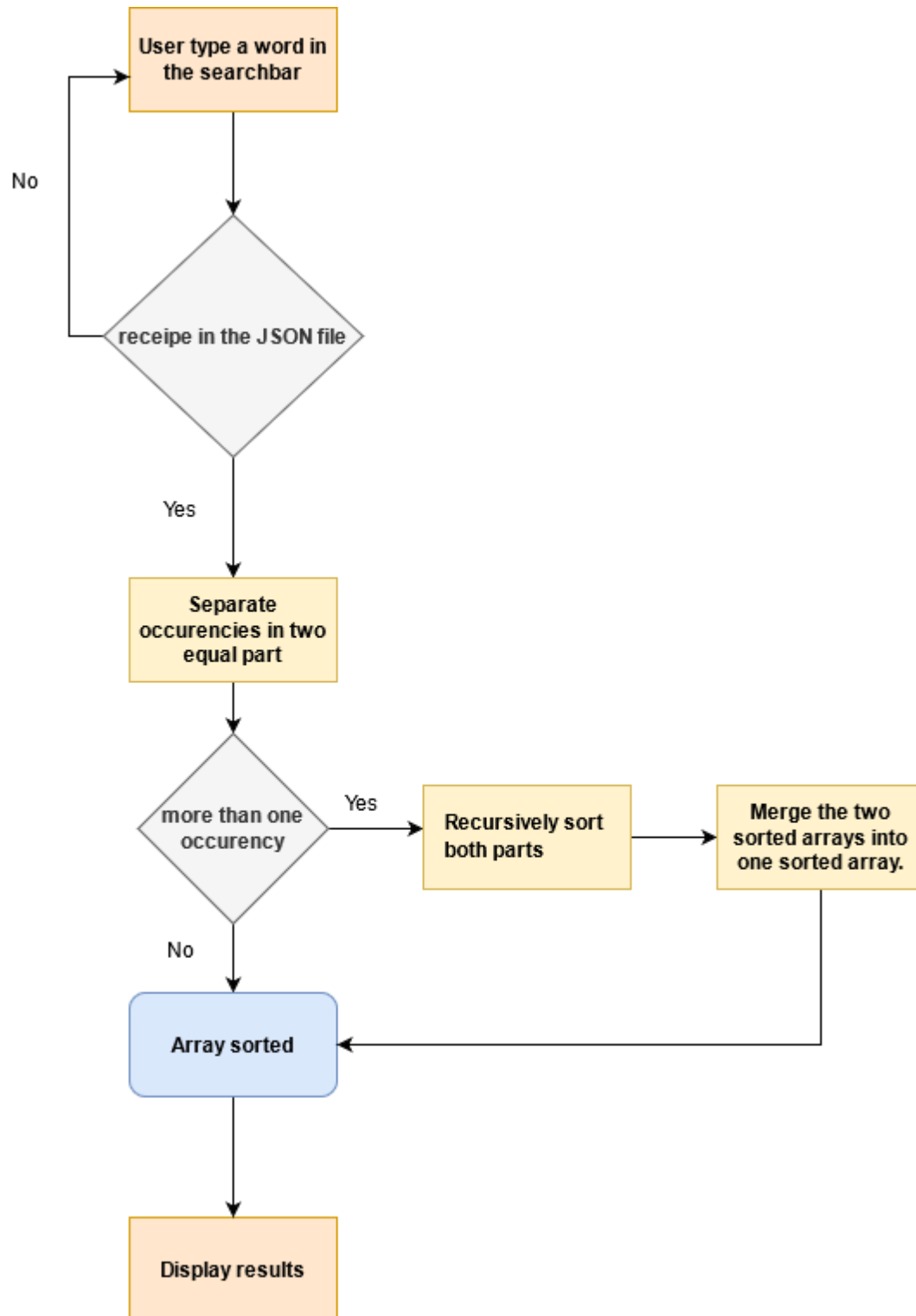


Figure 1 – Diagramme de l’algorithme tri-fusion

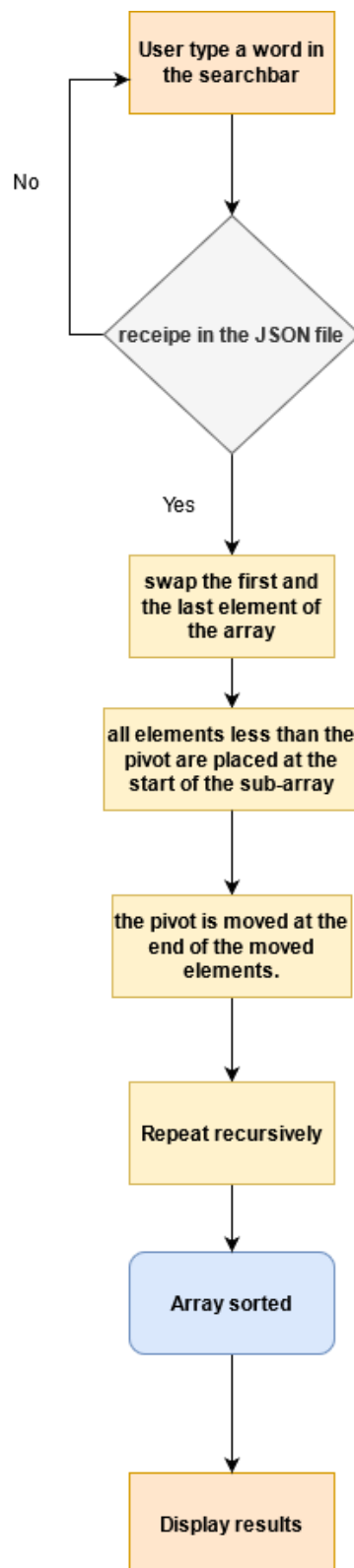


Figure 2 – Diagramme de l’algorithme tri rapide