

## Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche par mots clé en 2 étapes		Fonctionnalité #2
<b>Problématique</b> : Obtenir les meilleures performances possibles pour une recherche par mot clé, implémentée sur un champ de recherche principal, puis sur les résultats de cette recherche (recherche avancée)		
Option 1 : DETAILS DE L'IMPLEMENTATION : PREFIX TRIE		
DESCRIPTION DU 'TIME COMPLEXITY' de la recherche non linéaire (uniquement la recherche dans le trie, omettant la construction préalable)		
FUNCTION	BIG O 'n' representant la taille des nodes (qui est fonction du nbre de mots entrés dans le trie)	
<p>⇒ CHERCHER UN MOT dans le trie</p> <ul style="list-style-type: none"><li>► Split mot to array</li><li>► For Loop : chaque lettre du mot<ul style="list-style-type: none"><li>▷ <b>node has</b> letter ?</li><li>▷ <b>node get</b> letter</li></ul></li><li>► Go to last node (recursion)<ul style="list-style-type: none"><li>▷ node keys <b>For Loop</b></li><li>▷ suggestions array <b>includes</b> recipe ?</li><li>▷ suggestions <b>push</b> recipe</li></ul></li></ul>	<p>⇒ O(n)</p> <p>⇒ O(n)</p> <p>⇒ O(1)</p> <p>⇒ O(1)</p> <p>⇒ O(n)</p> <p>⇒ O(n)</p> <p>⇒ O(1)</p>	
<p>⇒ TOTAL</p>	<p>⇒ 4 opérations de O(n)</p> <p>⇒ + 3 opérations de O(1)</p> <p>⇒ <b>O(n)</b></p>	
<p>Conclusion : la recherche non linéaire se traduit bien par une complexité de O(n).</p> <p>Pour une évaluation plus précise, il faudrait évaluer également les performances liées à la construction initiale du trie.</p>		