

		<p>пунктами с помощью алгоритма Дейкстры</p> <p>Оптимизировать построенный код: 1) по производительности, 2) по потреблению памяти</p> <p>Обосновать выбор структуры данных для хранения информации о ребрах и вершинах</p>		
4.	Использование хеширования в задачах поиска в строке	<p>Для поиска в тексте необходимо применить реализации алгоритма на основе хеширования.</p> <p>Сравнить скорость работы алгоритма с</p> <ul style="list-style-type: none"> Алгоритмом Боуера – Мура и его модификациями; Алгоритмом Кнута-Морриса-Пратта; 	<p>Проиллюстрировать работу реализаций алгоритма привести оценки их производительности.</p> <p>При выводе информации о результатах поиска либо подсветить фрагмент текста, внутри которого найдено слово (несколько слов перед и после найденного); либо вывести этот фрагмент</p> <p>При выводе информации о</p>	<p>https://www.youtube.com/watch?v=eFuMs6Lc9S0&list=PLJOzdkh8T5koEPv-R5W0ovmL_T2BjB1HX&index=6</p> <p>Кормен, часть III Топп, Форд, глава 14</p> <p>Кормен, часть VI Топп, Форд. Глава 14</p> <p>https://neerc.ifmo.ru/wiki/index.php?title=%D0%9F%D0%BE%D0%B8%D1%81%D0%BA_%D0%BF%D0%BE%D0%B4%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8_%D0%B2_%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B5_%D1%81_%D0%B8%D1%81%D0%BF%D0%BE%D0%B%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D0%BC_%D1%85%D0%B5%D1%88%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A0%D0%B0%D0%B1</p>

	<p>Ввести класс «Совпадение», содержащий информацию о том, где совпадение найдено в тексте, зафиксировано частичное или полное совпадение</p> <p>Реализовать процесс нечёткого поиска: часть букв в найденном слове может не совпадать с символами в искомом слове: “Text”, “testa” => совпадение - 60% или 75%.</p> <p>При встрече процента выше некоторого предела фиксируется совпадение</p> <p>Ввести функцию сортировки результатов по проценту совпадения</p>	<p>результатах нечёткого поиска, выводить процент совпадения</p>	<p>%D0%B8%D0%BD%D0%B0-%D0%9A%D0%B0%D1%80%D0%BF%D0%B0</p> <p>https://e-maxx.ru/algo/string_hashes</p> <p>https://algorithmica.org/ru/hashing</p>
--	--	--	--

		Оптимизировать построенный код: 1) по производительности, 2) по потреблению памяти		
5.	Моделирование и оптимизация дерева отрезков	<p>Реализовать класс «Дерево отрезков»</p> <p>Реализовать необходимые для класса функции: конструкторы, деструкторы, нужные методы</p> <p>Реализовать поиск значения функции на отрезке</p> <p>Реализовать групповое изменение значений в определенном отрезке</p> <p>Для выполнения групповых операций реализовать отложенное выполнение команд</p> <p>Оптимизировать построенный код: 1) по производительности, 2) по потреблению</p>	<p>Продемонстрировать работу дерева отрезков на определенных полуинтервалах</p> <p>Проиллюстрировать оценки структуры данных по памяти и производительности</p>	<p>https://www.youtube.com/watch?v=MmemGjxsZTc&list=PLJOzdkh8T5koEPv-R5W0ovmL_T2BjB1HX&index=10</p> <p>http://brestprog.by/topics/segmenttree/</p> <p>https://habr.com/ru/post/114980/</p> <p>https://habr.com/ru/post/115026/</p> <p>https://e-maxx.ru/algo/segment_tree</p> <p>https://proglib.io/p/algoritmy-i-struktury-dannyh-na-c-derevya-otrezkov-2019-10-11</p>