**Анализ**

Тази задача съчетава един стандартен алгоритъм за С група и малко по-нестандартен за тази по-малка група. Предвидени са точки за частични решения във формата на подзадачи, като така е намалено и случайното хващане на точки.

Първата подзадача е за 30 точки. Главното, което трябва да се види в задачата, е, че ако знаем отговора, задачата се улеснява многократно. Знаейки предполагаем отговор, единствено трябва да видим на кои позиции се среща търсената дума (като може да има припокривания) и почваме да броим от първото срещане дали ще можем да намерим поне ***K*** индекса, при които съседните (подредени по големина, разбира се) са на по-голямо или равно разстояние от фиксирания отговор. За тази подзадача се очаква да се пусне пълне изчерпване за търсеният отговор (т.е. да се проверят всички възможности). Тук не е предвидено по-добро реализиране на проверката на кои позиции думата се среща в текста. Сложността е .

Втората подзадача е за 30 точки. Сега трябва да се направи най-важната стъпка в задачата – идеята за двоично търсене по отговора. Би трябвало интуитивно на всички да е ясно, че ако за дадено разстояние успеем да намерим поне ***K*** индекса, то за по-малко със сигурност ще можем и трябва да видим дали има по-голямо. Обратно – ако не успеем за дадено, за по-големи със сигурност няма да успеем, но е възможно за по-малки. Тези разсъждения показва, че функцията на отговора е монотонна. Затова ще реализираме двоично търсене по исканият отговор. Това доста подобрява сложността и решението става: .

Третата подзадача е за 20 точки. Следващата важна стъпка е да видим, че няма смисъл всеки път да намираме кои са индексите на срещанията на думата в текста, а това можем да го направим преди самото двоично търсене. Така за всяка итерация на търсенето просто обхождаме генерирания масив с позициите на срещания на думата. Сложността на това решение е .

Последната подзадача също е за 20 точки. Тук ще подобрим частта с намирането на срещанията на думата. Логичен въпрос, който трябва да си зададат състезателите, е защо има само две букви в текста. Това би трябвало да ги насочи да си мислят за думата като двоично число и кодиране. Именно така може да се подобри търсенето на конкретната дума. Понеже имаме много малка дължина на думата, то кодът ѝ като двоично число ще се побере в типа . Доста лесно можем да минем през текста за всяка позиция да видим кода на думата, образувана от следващите ***M*** букви в текста, и да проверим дали е равен на този на думата на **Д**ени. Така вече константно намираме срещанията с едно обхождане на текста. Реално това е частен случай на по-сложната техника хеширане, използвана в по-горни групи. Тук улеснението идва от това, че дължината на думата е много малка и можем да имаме идеален случай на хеш – такъв без колизии. Окончателната сложност за задачата е .

*Автор: Илиян Йорданов*