

Reading

(решение)

Има два основни подхода, които могат да решат задачата. Единият от тях, за съжаление, би бил бавен при дадените ограничения, но за него са предвидени 50-60 точки при добра имплементация.

Първи подход: динамично оптимизиране. Повечето от състезателите веднага ще видят сравнително лесното решение на базата на динамично оптимизиране. Тъй като за дадена сума N и буква, която искаме да сложим L ние зависим единствено от това колко на брой думи имаме със сума $N - \text{factor}(L, A_i)$, където A_i е всяка друга възможна буква, динамична таблица $\text{dyn}[26][N]$ е напълно достатъчна за да намерим отговор на задачата. Тъй като за всяка клетка ние трябва да проверим всички останали 26 букви, общата сложност на това решение е $O(26 * 26 * N) = O(N)$. Това решение не хваща пълен брой точки, защото първо N е относително голямо число, второ константата $(26 * 26)$ е доста голяма.

Вторият подход включва малко по-нетривиални знания и досетливост. Всеки състезател в тази група (би трябвало да) знае, че броят пътища с ТОЧНО N стъпки в граф, зададен чрез матрица на съседство е тази матрица, вдигната на N -та степен. Можем да построим граф на връзките между съседни букви и да използваме неговата матрица за да намерим отговор за нашата задача. Въпреки, че N е голямо число, можем да се възползваме от бързото (логаритмично) вдигане на степен и това ще ни осигури достатъчно добър алгоритъм за да се вместим във времевите ограничения. Тук обаче идва проблем – какво правим с различните разстояния между буквите? Да сложим тях вместо единиците би било грешно (тогава алгоритъма не работи), за това трябва да помислим малко повече. Тъй като пътищата ни са сравнително къси (до 5), можем да разширим графа като добавим изкуствени върхове, които ни удължават пътищата между букви с цена повече от 1, като запазват свойството всяко ребро в графа да е с цена точно 1. Ако повдигнем новата матрица на N -та степен ще получим именно броя на думите, с дължина N .

Последният проблем, с който трябва да се справим, е това, че не ни интересуват само думите със сума ТОЧНО N , ами с по-малка или равна такава. Това се решава чрез добавяне на изкуствена буква, която можем да считаме за „празен“ символ – тоест тя не добавя нищо към думата. От нея ще направим ребра до всяка буква (всяка валидна дума трябва да започне все някога), както и ребро към самата себе си (за да можем да съкращаваме думите с произволна дължина).

Сложност: Тъй като за всеки връх ни трябва да го разширим в 5 нови (за да гарантираме, че можем да ползваме максималната цена) получаваме матрица с размер $A[26*5+1][26*5+1]$. Имаме максимум логаритъм от $1,000,000,000 = 30$ стъпки за вдигането на степен и $K^{\wedge}3$ алгоритъм за умножаване на матрици (където K е размерността на

матрицата, в случая 131). Общата сложност е $O(131^3 * \log N)$. 131^3 , макар и константа, е достатъчно голямо число да влияе значително на времето за изпълнение, така че не бива да бъде пропускано от състезателите.

Автор: Александър Георгиев