

Гирлянда из флажков

Если бы максимальная длина гирлянды-палиндрома была чётной, было бы достаточно отсортировать для каждой буквы все возможные стоимости её взять, затем брать для каждой буквы две минимальные стоимости, если они ещё не были взяты, и среди всех возможных букв выбирать такую минимальную пару. Можно продолжать так делать, пока не кончится бюджет, и вывести в качестве ответа те буквы, которые вошли в бюджет. Это будет оптимальный ответ, так как никогда не имеет смысла брать не самую дешёвую пару. Взятые пары мы расставим симметрично относительно центра гирлянды и получим палиндром.

Ответ не всегда бывает чётной длины, поэтому случай нечётной длины нужно рассмотреть отдельно. Для этого переберём букву, которая будет стоять в центре (от **a** до **z**), и сразу же возьмём флажок минимальной стоимости с ней. Далее осталось просто решить задачу для чётной длины палиндрома, убрав выбранный флажок и переразбив флажки для выбранного символа на пары (начиная со второго по стоимости флажка, а не с первого). Если наивно повторить решение для чётного случая, получится асимптотика $\mathcal{O}(C \cdot (n \log n))$, но его можно улучшить.

Заметим, что после применения решения для чётной длины палиндрома мы можем получить отсортированный по возрастанию стоимостей массив всех пар букв (пары получены описанным способом разбиения). Теперь будем так же перебирать букву, которая будет стоять в середине, и построим новый массив пар стоимостей для этой буквы, убрав из стоимостей минимальную (её мы ставим в середину). Отсортируем новые пары для текущей буквы, и теперь задача сводится к тому, чтобы получить из двух отсортированных массивов (массива для всех букв, который мы получили ещё до перебора буквы, и массива новых пар для буквы) один отсортированный. Это можно сделать алгоритмом слияния двух массивов за сумму их длин (поддерживая по указателю на текущее место в каждом массиве, и каждый раз добавляя в новый массив число, которое меньше, и двигая соответствующий указатель). Чтобы учесть, что в массиве для всех букв остались некорректные пары для текущей перебираемой буквы, можно просто пропускать все такие вхождения в массиве в очередной итерации алгоритма слияния. Также заметим, что на самом деле можно не создавать новый отсортированный массив, а сразу применять алгоритм, описанный для случая палиндрома чётной длины, только изначально нужно добавить к стоимости набираемой гирлянды букву, которая стоит в середине.

Сортировка всех букв вместе будет за $\mathcal{O}(n \log n)$, сортировки букв по отдельности также дадут $\mathcal{O}(n \log n)$, так как суммарно всех букв n , а перебор буквы и слияние произойдёт за асимптотику $\mathcal{O}(C \cdot n)$. Таким образом, получаем решение за $\mathcal{O}(n \cdot (\log n + C))$.