解决这个问题的关键观察是，如果字符串 $y_1y_2\dots y_n$ 的分割是一个最优分割，那么字符串 $y_1y_2\dots y_{n-1}$ 的分割也将是字符串 y 的前缀的最优分割，其中排除了 y_n (因为否则我们可以替换原问题中前缀的最优解并得到一个更好的解)。

基于这个观察，我们设计子问题如下。令 $\text{Opt}(i)$ 表示由 y 的前 i 个字符组成的前缀的最佳分割的分数。我们声称以下递推关系：

$$\text{Opt}(i) = \max_{j \le i}\{\text{Opt}(j-1) + \text{Quality}(j\dots n)\}$$

将给出正确的优化分割（其中 $\text{Quality}(\alpha\dots\beta)$ 表示由字符从起始位置 α 到结束位置 β 形成的单词的质量）。请注意，所需的解是 $\text{Opt}(n)$。

我们用关于索引 i 的归纳法证明上述公式的正确性。基本情况是简单的，因为只有一个字母的单词只有一个字符。

对于归纳步骤，假设我们知道上述的 Opt 函数对于小于 i 的索引给出最优解，并且我们想证明 $\text{Opt}(i)$ 的值是前缀 y 到第 i 个字符的任何分割的最优质量。考虑此前缀的最优分割中的最后一个单词。假设它以索引 $j \le i$ 结束。然后根据我们的关键观察，只包含前 $j-1$ 个字符的前缀也必须是最优的。但是根据我们的归纳假设，$\text{Opt}(j)$ 将给出上述最优分割的值。因此最优质量 $\text{Opt}(i)$ 将等于 $\text{Opt}(j)$ 加上最后一个单词的质量。

但请注意，我们的上述递推关系恰好计算了最后一个单词的每种可能性的这种计算。因此，我们的递推关系将正确地找到最优分割的质量。

对于运行时间，一个简单的实现（对上述公式的直接评估）从索引 1 开始直到 n，其中 n 是输入字符串中的字符数，将产生一个二次算法。