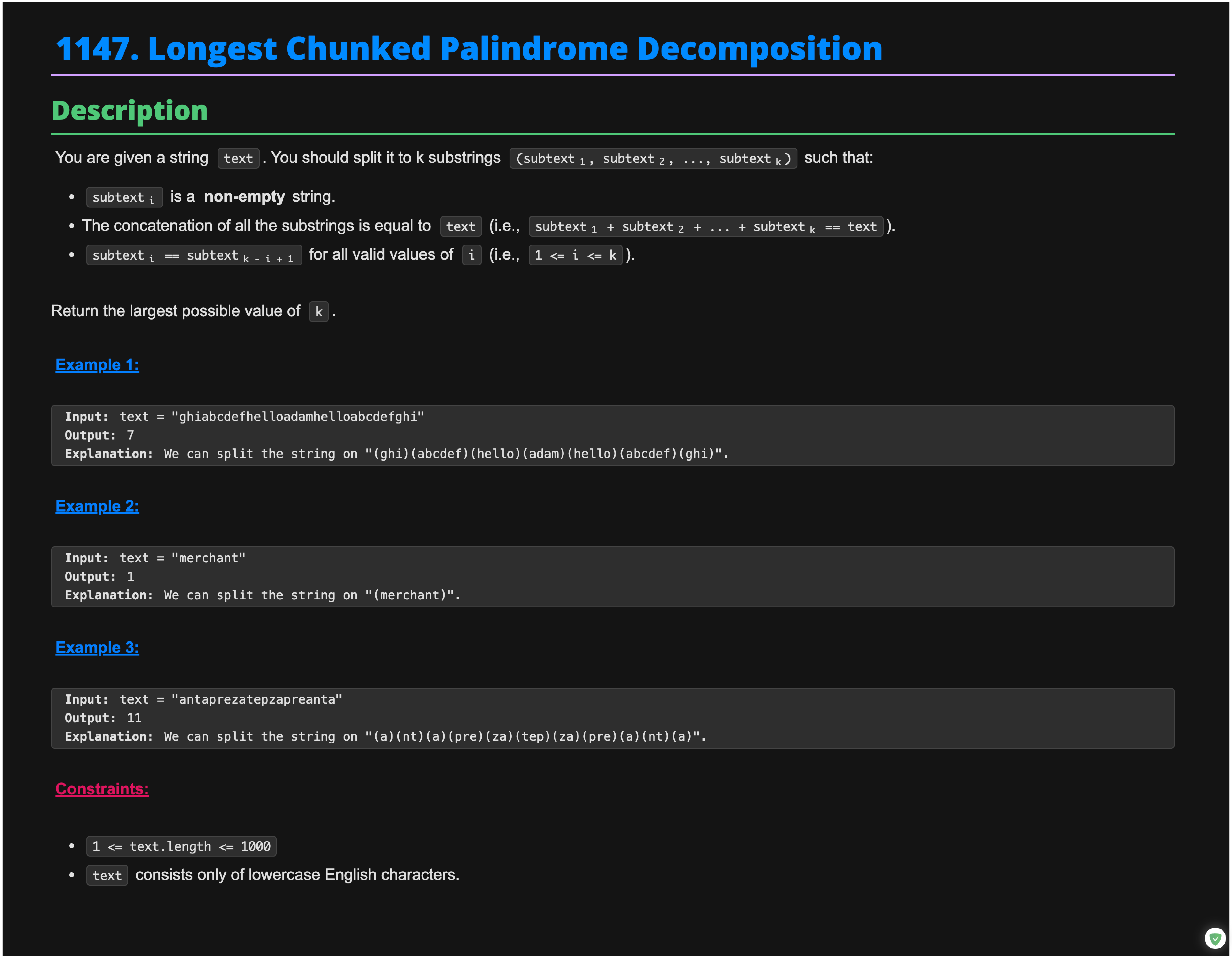

1147. Longest Chunked Palindrome Decomposition

Description

You are given a string `text`. You should split it to k substrings $(\texttt{subtext}_1, \texttt{subtext}_2, \dots, \texttt{subtext}_k)$ such that:

- $\texttt{subtext}_i$ is a **non-empty** string.
- The concatenation of all the substrings is equal to `text` (i.e., $\texttt{subtext}_1 + \texttt{subtext}_2 + \dots + \texttt{subtext}_k == \texttt{text}$).
- $\texttt{subtext}_i == \texttt{subtext}_{k - i + 1}$ for all valid values of `i` (i.e., `1 <= i <= k`).

Return the largest possible value of `k`.

Example 1:

```
Input: text = "ghiabcdefhelloadamhelloabcdefghi"
Output: 7
Explanation: We can split the string on "(ghi)(abcdef)(hello)(adam)(hello)(abcdef)(ghi)".
```

Example 2:

```
Input: text = "merchant"
Output: 1
Explanation: We can split the string on "(merchant)".
```

Example 3:

```
Input: text = "antaprezatepzapreanta"
Output: 11
Explanation: We can split the string on "(a)(nt)(a)(pre)(za)(tep)(za)(pre)(a)(nt)(a)".
```

Constraints:

- `1 <= text.length <= 1000`
- `text` consists only of lowercase English characters.