9Q Construct the Partial Suffix Array of a String

Partial Suffix Array Construction Problem
Construct the partial suffix array of a string.

Input: A string *Text* and a positive integer k.
Output: SUFFIXARRAY$_k$(*Text*), in the form of a list of ordered pairs (i, SUFFIXARRAY(i)) for all nonempty entries in the partial suffix array.

```
7  $
1  ANANAS$
3  ANAS$
5  AS$
0  BANANAS$
2  NANAS$
4  NAS$
6  S$
```

Formatting

Input: A string *Text* and a positive integer k.
Output: A newline-separated list of space-separated ordered pairs (i, SUFFIXARRAY(i)) for all nonempty entries in SUFFIXARRAY$_k$(*Text*).

Constraints

- The length of *Text* will be between 1 and 10^5.
- The integer k will be between 1 and 10^1.

Test Cases

Case 1

Description: The sample dataset is not actually run on your code.

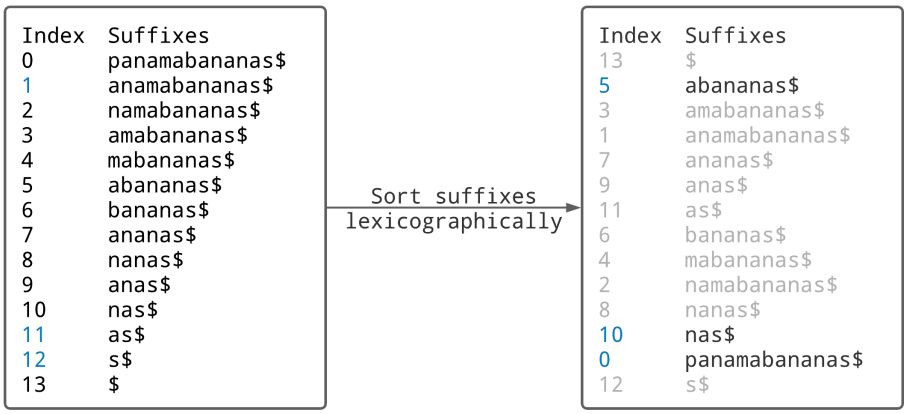
Input:

```
panamabananas$
5
```

Output:

```
1 5
11 10
12 0
```

Figure:



Shown above is a general (and inefficient) construction of the partial suffix array of the input string `panamabananas` with $k = 5$. We first generate all suffixes of *Text* before sorting the suffixes lexicographically and outputting the indices representing the sorted suffixes as the complete suffix array of *Text*. Finally, we output only the indices divisible by $k = 5$.

Case 2

Description: There are repeats in *Text*.

Input:

```
AATCAATC$
4
```

Output:

```
0 8
1 4
2 0
```

Case 3

Description: There are no repeats in *Text*.

Input:

```
ATCG$
3
```

Output:

```
1 0
3 3
```

Case 4

Description: Large regions of *Text* being a single character or short tandem repeat (STR).

Input:

```
AAACA$
5
```

Output:

```
0 5
2 0
```

Case 5

Description: Many different characters in one pattern.

Input:

```
ABCFED$
3
```

Output:

```
0 6
1 0
6 3
```

Case 6

Description: A larger dataset of the same size as that provided by the randomized autograder. Check input/output folders for this dataset.