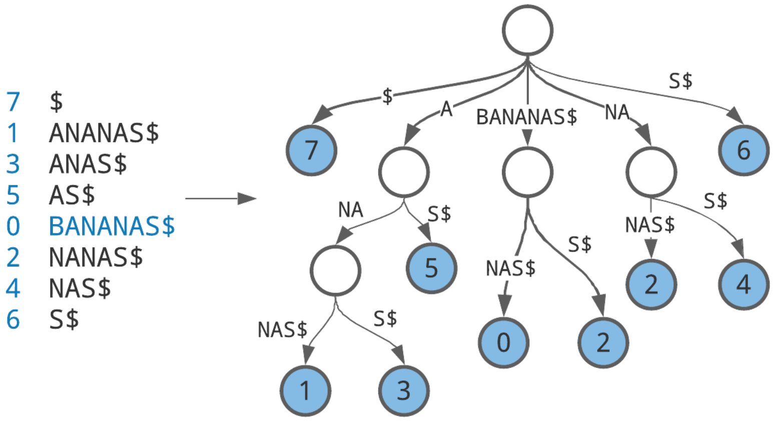
9R Construct a Suffix Tree from a Suffix Array

Suffix Tree Construction from Suffix Array Problem
Construct a suffix tree from the suffix array and LCP array of a string.

Input: A string *Text*, SUFFIXARRAY(*Text*), LCP(*Text*).
Output: The strings labeling the edges of SUFFIXTREE(*Text*), in any order.



Formatting

Input: A string *Text*, followed by SUFFIXARRAY(*Text*), followed by *LCP*(*Text*).
Output: A space-separated list of edge labels from the constructed suffix tree (in any order).

Constraints

- The length of *Text* will be between 1 and 10^3.
- The length of SUFFIXARRAY(*Text*) will be between 1 and 10^3.
- The length of LCP(*Text*) will be between 1 and 10^3.

Test Cases

Case 1

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

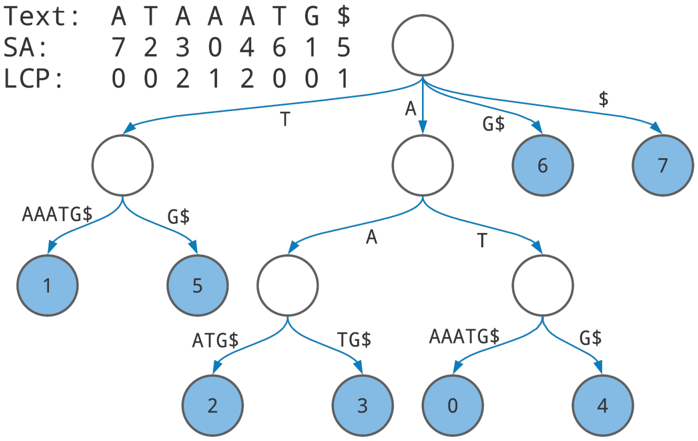
Input:

```
ATAAATG$
7 2 3 0 4 6 1 5
0 0 2 1 2 0 0 1
```

Output:

```
$ $ A A AAATG$ AAATG$ ATG G$ G$ G$ T T TG$
```

Figure:



Above is the suffix tree for the string ATAAATG$ (notice the $ appended to the end of our input string ATAAATG). Each path from the root to each of the leaves (shown in blue) represents the suffix of ATAAATG$ corresponding to the index in the leaf.

Case 2

Description: There are repeats in *Text*.

Input:

```
AATCAATC$
8 4 0 5 1 7 3 6 2
0 0 4 1 3 0 1 0 2
```

Output:

```
$ $ $ $ A AATC$ AATC$ AATC$ ATC C TC TC
```

Case 3

Description: There are no repeats in *Text*.

Input:

```
ATCG$
4 0 2 3 1
0 0 0 0 0
```

Output:

```
$ ATCG$ CG$ G$ TCG$
```

Case 4

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

Input:

```
AAACA$
5 4 0 1 2 3
0 0 1 2 1 0
```

Output:

```
$ $ A A ACA$ CA$ CA$ CA$
```

Case 5

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