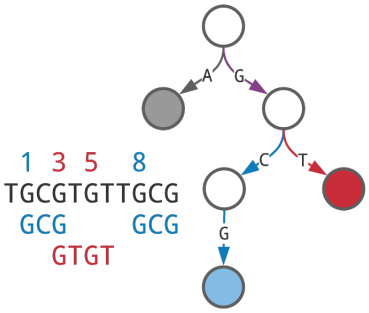
9B Implement TrieMatching

Trie Matching Problem
Find all occurrence of a collection of patterns in a text.

Input: A string *Text* and a collection of (shorter) strings *Patterns*.
Output: Each string *Pattern* in *Patterns* followed by the starting positions in *Text* where *Pattern* appears as a substring.



Formatting

Input: A string *Text* and a space-separated list of strings *Patterns*.
Output: Each string *Pattern* in *Patterns* followed by the starting positions in *Text* where *Pattern* appears as a substring.

Constraints

- The length of *Text* will be between 1 and 10^4.
- The number of patterns in the string-set *Patterns* will be between 1 and 10^1.
- The length of any one pattern in *Patterns* will be between 1 and 10^4.

Test Cases

Case 1

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

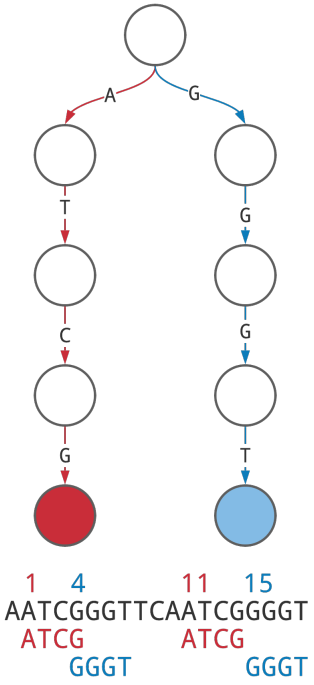
Input:

```
AATCGGGTTCAATCGGGGT
ATCG GGGT
```

Output:

```
ATCG: 1 11
GGGT: 4 14
```

Figure:



Above is the trie for `ATCG` and `GGGT` (shown in red and blue respectively). Shown below the trie is our input string *Text* as well as a representation of where our *Patterns* appear in *Text*.

Case 2

Description: There is no match for any *Pattern* in *Text*.

Input:
```
AATCGGGTTCAATCGGGGT
GGGA
```

Output:
```
GGGA:
```

Case 3

Description: A *Pattern* in *Patterns* is identical to *Text*.

Input:
```
AATC
AATC
```

Output:
```
AATC: 0
```

Case 4

Description: Patterns with repeats or palindromic sequences, overlapping occurrences of a pattern, and/or incomplete matches at *any* point in *Text*.

Input:
```
ATATATA
ATA TAT
```

Output:
```
ATA: 0 2 4
TAT: 1 3
```

Case 5

Description: Matches that only occur once (beginning and end as well).

Input:
```
GAGCAT
GAG
```

Output:
```
GAG: 0
```

Case 6

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