

2B Find a Median String

Median String Problem

Find a median string.

Input: A collection of strings Dna and an integer k .

Output: A k -mer $Pattern$ that minimizes $D(Pattern, Dna)$ among all possible choices of k -mers.

```
CTTAAC
GATATC
ACGGCG
CTAAAG
```

AAA

Formatting

Input: An integer k , followed by a newline-separated collection of strings Dna .

Output: A string representing a k -mer $Pattern$ that minimizes $d(Pattern, Dna)$ over all k -mers $Pattern$ (If multiple answers exist, you may return any one).

Constraints

- The integer k will be between 1 and 10^1 .
- The number of strings in Dna will be between 1 and 10^2 .
- The length of each string in Dna will be between 1 and 10^2 .
- Each string in Dna will be a DNA string.

Test Cases

Case 1

Description: The sample dataset is not actually run on your code. Notice that there are technically two solutions to the problem (ACG and GAC are equally optimal), but the problem specifically states to only return a single output (you can arbitrarily pick any optimal solution).

Input:

```
3
AAATTGACGCAT GACGACCACGTT CGTCAGCGCCTG GCTGAGCACCGG AGTACGGGACAG
```

Output:

```
ACG
```

Case 2

Description: This dataset checks that your output is the correct length. Notice that there are technically two solutions to the problem (ACG and CGT are equally optimal), but the problem specifically states to only return a single output (you can arbitrarily pick any optimal solution). Also, since $k = 3$ in this dataset, we check that your output is exactly of length k (should not be any longer or shorter than this).

Input:

```
3
ACGT ACGT ACGT
```

Output:

```
ACG
```

Case 3

Description: This dataset checks if your code considers k -mers that do not occur in *Dna*. Notice that the best 3-mer is AAA, which does not actually occur in any of the sequences in *Dna*. It is perfectly fine that our optimal median string does not actually occur in any of the sequences in *Dna* (similar to the Frequent Words With Mismatches Problem from chapter one).

Input:

```
3
ATA ACA AGA AAT AAC
```

Output:

```
AAA
```

Case 4

Description: This dataset checks that your output only contains a single k -mer. Notice that there are technically two solutions to the problem (AAG and AAT are equally optimal), but the problem specifically states to only return a single output (you can arbitrarily pick any optimal solution).

Input:

3

AAG AAT

Output:

AAG

Case 5

Description: A larger dataset of the same size as that provided by the randomized autograder.