Counting k-mers

Welcome to Generate's Tech Challenge!

Thanks again for applying to be a Software Engineer at Generate! We're introducing a technical challenge this semester that we hope is interesting. This challenge should take about an hour - we're asking you to:

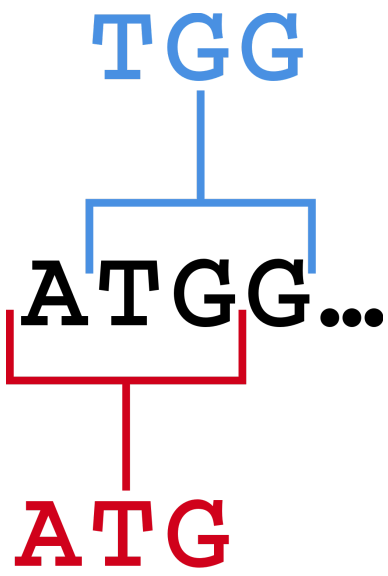
1. Make an HTTP request to get some data
2. Perform some string manipulation
3. Make an HTTP request to submit your solution.

If you get the right answer, you should see the following text as a response: `'Correct - nice work!'`

Notice that we're not specifying a language! Use whatever language you're comfortable with.

What are k-mers?

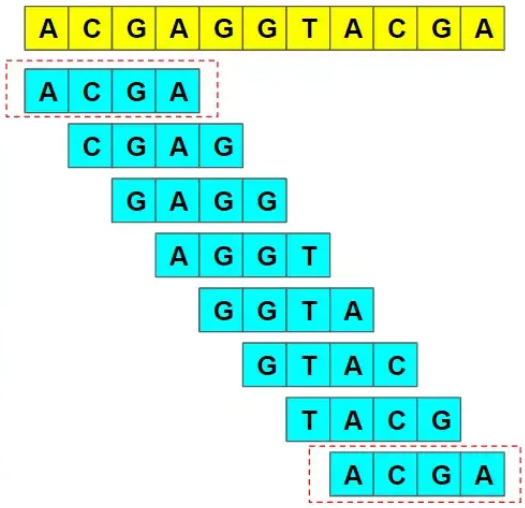
In computational biology, *k-mers* are substrings of length *k* that are contained within some sort of biological sequence.



The sequence ‘ATGG’ has two 3-mers: ‘ATG’, and ‘TGG’

In fields such as computational genomics, where the sequences in question can be incredibly large, it’s often useful to consider the count of all the k-mers instead of performing computations over the entire string.

	Total	Distinct	Unique
ACGA	2	1	0
CGAG	1	1	1
GAGG	1	1	1
AGGT	1	1	1
GGTA	1	1	1
GTAC	1	1	1
TACG	1	1	1



In the example above, we represent the string `ACGAGGTACGA` by counting its 4-mers

```
{
  "ACGA": 2,
  "CGAG": 1,
  "GAGG": 1,
  "AGGT": 1,
  "GGTA": 1,
  "GTAC": 1,
  "TACG": 1
}
```

Your Challenge

Write a function that, given a string of arbitrary length, counts all the 3-mers. Submit a solution to `generate-tech-app.xyz/submit/{{your token}}` in the format described above.

For example, a correct response to the challenge string `aabbceedeaab` would be

```
{
    "aab": 2,
    "abb": 1,
    "bbc": 1,
    "bce": 1,
    "cee": 1,
    "eed": 1,
    "ede": 1,
    "dea": 1,
    "eaa": 1
}
```

The challenge is hosted at `generate-tech-app.xyz` - here are the endpoints you'll need to use:

Endpoints

RegisterUser:

> **This is the first thing you should do** - use this method to register for the tech challenge! You'll receive your token and the challenge string you'll be working with.
>
> You can hit this endpoint through your command line with something like
>
> ```
> curl -X POST https://generate-tech-app.xyz/register -H 'Content-Type: application/json'
>  -d '{"name": "Your name", "nuid": "your nuid"}'
> ```
>
> **Method**: Post
>
> **Route**: `generate-tech-app.xyz/register`
>
> **Body**:
>
> ```
> {
>     "name": "Your name here",
>     "nuid": "Your nuid (as a string)"
> }
> ```

Response:

```
{
    "token": "Your token - save this!",
    "challenge_string": "Your challenge string - save this too!"
}
```

ForgotToken

Use this endpoint to get back your token if you lose it

Method: Get

Route: `generate-tech-app.xyz/forgot_token/{{your nuid}}`

Body: There's no request body

Response:

```
{
    "token": "here's your token back"
}
```

GetChallengeString

Use this endpoint to get back your challenge string if you lose it

Method: Get

Route: `generate-tech-app.xyz/challenge/{{your token}}`

Body: There's no request body

Response:

```
{
    "challenge_string": "Here's your challenge string!"
}
```

SubmitSolution

Use this endpoint to submit your solution

Method: Post

Route: `generate-tech-app.xyz/submit/{{your token}}`

Body:

```
{
  "AAA": 3,
  "ABC": 2,
  ...
  // Obviously this is an example, but this is how it should be formatted
}
```

Response: The response varies based on if the solution you submit is correct or not. You'll either get `'Correct - nice work!'` or `'Incorrect Solution'`

How do I get started?

1. Register for the challenge
2. Find and get comfortable with an HTTP library in whatever language you've chosen
3. Count all of the **3-mers** for the challenge you received in step 1.
4. Send that back to us using the submission endpoint!

Good Luck!