

Algorithm exercise (estimated time: 72 minutes)

Write an algorithm with the following signature:

```
int findPattern(String text, String word)
```

The first parameter is a text containing words. The words are separated by one blank character (space). The second parameter contains one word. All words can only contain small letters in the English alphabet (a-z). The same letter *cannot* occur more than once in `word`.

The task is to find words in the parameter `text` that contain all the letters in the parameter `word`.

The return value is an integer and can have the following values:

1. There is a word in `text`, which is identical with `word`.
2. There is a word in `text`, which contains the same letters as in `word`, but in a different order, e.g., `word` is equal to "som", word in `text` equal to "mos".
3. There is a word in `text`, which contains all the letters in `word` and in the same order, e.g., `word` is equal to "som" and word in `text` equal to "storkemor", or in a different order, e.g., `word` equal to "som" and word in `text` equal to "ostemad".

If none of the criteria are fulfilled, the return value is 0.

More than one criterion can be fulfilled and the criterion with the lowest return value has precedence. That means that if there is an identical word the return value *must* be 1. If there is not an identical word but one with the same letters, the return value *must* be 2 etc.

The only methods from the `String/string`-classes you may use are `charAt()`, `length()`, `compareTo()` in Java and `at()`, `length()`, `compare()` in C++.

To receive maximum points, the best-case time complexity must be less than $O(\log N)$, and the parameter `text` may only be traversed once.

Øvelse i algoritmeskrivning (estimeret tidsforbrug: 72 minutter)

Skriv en algoritme med følgende signatur:

```
int findPattern(String text, String word)
```

Den første parameter indeholder en tekst bestående af ord. Ordene er adskilt af én blank karakter (mellemrum). Den anden parameter består af et enkelt ord. Alle ord kan kun bestå af små bogstaver i det engelske alfabet (a-z). Det samme bogstav kan *ikke* forekomme flere gange i `word`.

Opgaven går ud på at finde ord i parameteren `text`, som indeholder alle bogstaverne i parameteren `word`.

Returværdien er et heltal (integer) og kan antage følgende værdier:

1. Der findes et ord i `text`, som er identisk med `word`.
2. Der findes et ord i `text`, som indeholder de samme bogstaver som i `word`, men i en anden rækkefølge, fx `word` er lig "som", ord i `text` lig med "mos".
3. Der findes et ord i `text`, som indeholder samtlige bogstaver i `word` og i samme rækkefølge, fx `word` lig med "som" og ord i `text` lig med "storkemor", eller i en anden rækkefølge, fx `word` lig med "som" og ord i `text` lig med "ostemad".

Hvis ingen af kriterierne er opfyldt, returneres 0.

Flere af betingelserne kan være opfyldt, og kriteriet med den laveste returnærdi har præcedens. Dvs. at hvis der findes et identisk ord, så skal returnærdien være 1. Hvis der ikke findes et identisk ord, men et med de samme bogstaver, returneres 2 etc.

De eneste af `String/string`-klassens metoder, du må benytte dig af er `charAt()`, `length()`, `compareTo()` i Java og `at()`, `length()`, `compare()` i C++.

For at opnå maksimumpoints skal din algoritmes *best case* tidskompleksitet være mindre end $O(\log N)$, og parameteren `text` må kun gennemløbes én gang.