

Scratchによる アルゴリズム入門

第3回 文字列検索

2017.03.31

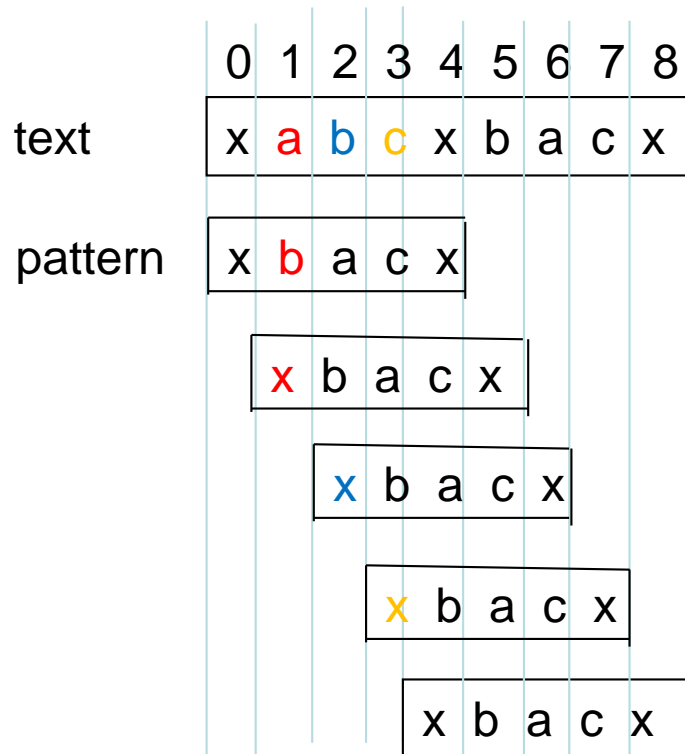
鎌倉シチズンネット(KCN)

文字列検索(1)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索

☆力まかせ法

文字列の先頭から順に照合して行き、文字列が一致しないことが分かったら、常に検索対象文字列の次の位置から比較を再開する、という方法である。



← 検索対象文字列

aとbが不一致なので1つ右にずらす

aとxが不一致なので1つ右にずらす

bとxが不一致なので1つ右にずらす

cとxが不一致なので1つ右にずらす

テキストとパターンが一致

文字列検索(2)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索

力まかせ法による文字列検索のアルゴリズム

変数の意味

text	検索対象文字列(検索される文字列)
pattern	検索文字列(検索する文字列)
textPos	text 中の位置
patternPos	pattern 中の位置
answer	文字列が見つかったtext 中の位置 (0 だと見つからなかった)

文字列検索(3)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索

力まかせ法による文字列検索のアルゴリズム

textPos を1に初期化する

answer が 0 でなくなるまで、または textPos が (テキスト の長さ - パターン の長さ + 1) より大きくなるまで以下の処理を繰り返す

 patternPos を1 に初期化する

 patternPos がパターンの長さより大きくなるか、または、テキストの (textPos + patternPos - 1) 番目の文字とパターンのpatternPos番目の文字が等しくなくなるまで、以下の処理を行う。

 patternPos を 1 ふやす。

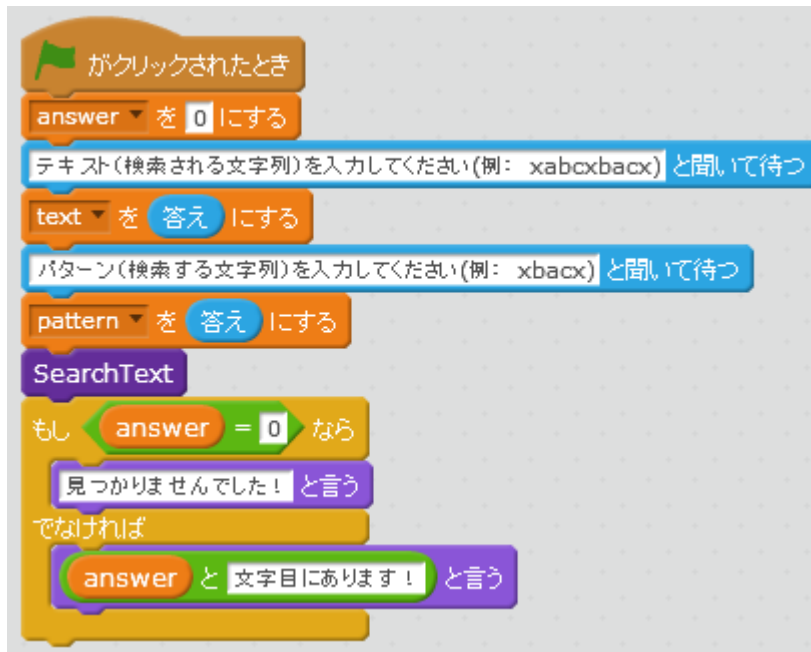
 もしも、patternPos の値がパターンの長さより大きければ、パターンが見つかったことになるので、以下の処理を行う。

 変数 answer に見つかった位置 textPosを設定する。

 textPos を 1 ふやす

文字列検索(4)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索



文字列検索(5)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索

```
定義 SearchText
textPos を 1 にする
answer = 0 ではない または textPos > text の長さ - pattern の長さ + 1 まで繰り返す
patternPos を 1 にする
patternPos > pattern の長さ または textPos + patternPos - 1 番目 ( text ) の文字 = patternPos 番目 ( pattern ) の文字 ではない まで繰り返す
patternPos を 1 ずつ変える
もし patternPos > pattern の長さ なら
answer を textPos にする
textPos を 1 ずつ変える
```

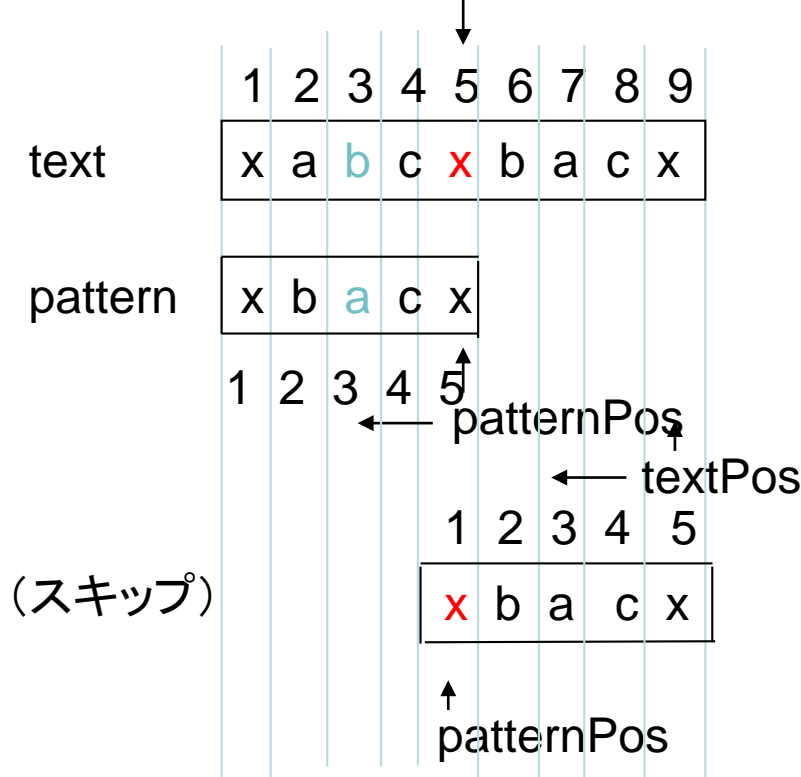
下記のプロジェクトをリミックスし、text 中にpattern が複数個存在する場合を考慮してください。

<https://scratch.mit.edu/projects/153376250/>

文字列検索(6)

- [アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索](#)

☆ [Boyer-Moore](#)法 (簡略版) による検索 (パターンの最後の文字から先頭に向かって照合する) ← textPos



textのtextPos番目と patternのpatternPos番目が等しい間、先頭に向かって比較する。

不一致になったら、textPos を元の位置 (照合を始めた位置: pstart) から、元の位置の文字によって決まるスキップ値 (左の例では 4) だけ右に進める。

スキップテーブル

a	b	c	x	その他の文字
2	3	1	4	5

patternPos が1 で文字が等しいときは照合が成功したことになる。

文字列検索(7)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索

Boyer-Moore法(簡易版)による文字列検索のアルゴリズム

変数の意味

text	検索対象文字列(検索される文字列)
pattern	検索文字列(検索する文字列)
textPos	text 中の位置
textStart	text 中の照合開始位置
patternPos	pattern 中の位置
answer	文字列が見つかったtext 中の位置 (0 だと見つからなかった)
skipTable	スキップテーブル(配列(リスト))

文字列検索(8)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索

Boyer-Moore法(簡易版)による文字列検索のアルゴリズム

テキストの照合開始位置をパターンの末尾に合わせる

テキストの照合開始位置がテキストの末尾を越えない間、以下を繰り返す

テキストの照合中の位置を照合開始位置に合わせる

パターンの照合中の位置をパターンの末尾に合わせる

テキストの textPos 番目の文字とパターンの patternPos 番目の文字 が等しい間、以下の処理を行う。

もしも、patternPosの値が 0 なら、パターンが見つかったことになるので、以下の処理を行う。

変数 answer に見つかった位置 textPosを設定する。

このスクリプトを止める。

textPos を 1 減らす。

patternPos を 1 減らす。

textStart をテキストの照合開始位置の文字に対応するスキップ値 skipNo だけ進める

文字列検索(9)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索

The image shows a Scratch script for a string search algorithm. The script is divided into two main sections: initialization and the search loop.

Initialization Section:

- When clicked, set `answer` to 0.
- Ask the user for a text to search (e.g., `xabcxback`).
- Set `text` to the user's answer.
- Ask the user for a pattern to search (e.g., `xback`).
- Set `pattern` to the user's answer.
- Define a function `BoyerMoore`.

BoyerMoore Function:

- Set `skipNo` to the length of `pattern`.
- Set `i` to 1.
- Repeat the following loop `pattern` length - 1 times:

 - If the character at index `i` of `pattern` matches the character at index `str`, set `skipNo` to `pattern` length - `i`.
 - Increment `i` by 1.

Final Output:

- If `answer` is 0, say "見つかりませんでした!" (Not found!).
- Otherwise, say "`answer` と 文字目にあります!" (Found in the character!).

(続く)

文字列検索(10)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索



上記のプロジェクトをリミックスして、未完成部分を完成させてください。

<https://scratch.mit.edu/projects/153487825/>

文字列検索(11)

- アルゴリズム入門 > 7章 文字列 > 7-7 文字列の検索

[解答例]

力まかせ法

<https://scratch.mit.edu/projects/153422409/>

Boyer-Moor 法

<https://scratch.mit.edu/projects/153429024/>