

# Scratchによる アルゴリズム入門

## 第5回 バックトラック

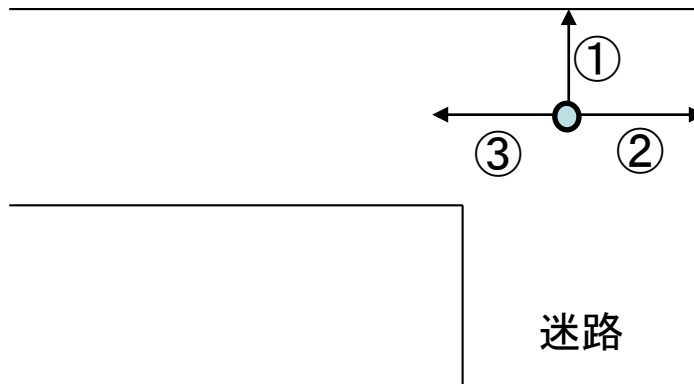
2017.04.02

鎌倉シチズンネット(KCN)

# バックトラック(1)

- バックトラックとは

バックトラックは「迷路」の探索などによく用いられるアルゴリズムで、やってみてダメだと分かったら元の位置(状態)まで戻って、別の道(選択肢)を選ぶ方法です。バックトラックは元の位置(状態)に戻るために、「状態」を保存しておく必要があります。



- ① ダメだ
- ② ダメだ
- ③ じゃ、こっち

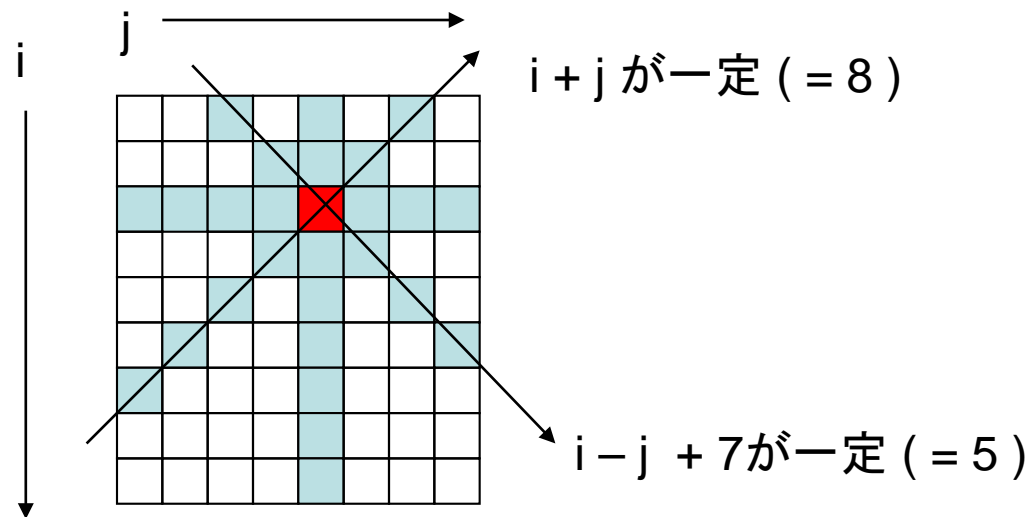
Scratchで作った「迷路」はこちらにあります。

<https://scratch.mit.edu/projects/87928065/>

# バックトラック(2)

- 8王妃(エイトクイーン)問題

8王妃問題とは、チェスの盤面上(8 × 8)に8つの王妃(クイーン)をお互いにとられないように配置する問題である。チェスのクイーンは将棋の飛車と角を合わせた動きをする(縦、横、対角線上を動く)。



Scratchで遊んでみよう。

<https://scratch.mit.edu/projects/89210088/>

# バックトラック(3)

- 8王妃問題(再帰版)

The image shows two parts of a Scratch script for the 8 Queens problem. The left part is a 'when clicked' event handler that sets up the board and starts a recursive function. The right part is the definition of the 'printQueen' function.

**Left Panel (Main Script):**

- がクリックされたとき
- すべて 番目を x から削除する
- すべて 番目を a から削除する
- すべて 番目を b から削除する
- すべて 番目を c から削除する
- すべて 番目を j から削除する
- 8 回繰り返す
  - 0 を x に追加する
- 8 回繰り返す
  - 0 を a に追加する
- 15 回繰り返す
  - 0 を b に追加する
- 15 回繰り返す
  - 0 を c に追加する
- answerNum を 0 にする
- queen 1

**Right Panel (Function Definition):**

- 定義 printQueen
- answerNum を 1 ずつ変える
- x は王妃の列位置を示しています。もっと見ますか? yes or no と聞いて待つ
- もし 答え = no なら
  - すべて を止める

再帰を用いた8王妃のプロジェクトです。これを、再帰を用いずに、繰り返しを使うようにリミックスしてください。

<https://scratch.mit.edu/projects/153684874/>

(次頁に続く)

# バックトラック(4)

- 8王妃問題(再帰版)

```
定数 queen i
1 を 最後 番目に挿入する(j)
最後 番目(j) < 8 でない まで繰り返す
もし 最後 番目(j) 番目(a) = c かつ i + 最後 番目(j) - 1 番目(b) = c なら
もし i - 最後 番目(j) + 8 番目(c) = c なら
i 番目(x) を 最後 番目(j) で置を換える
もし i < 8 なら
最後 番目(j) 番目(a) を 1 で置を換える
i + 最後 番目(j) - 1 番目(b) を 1 で置を換える
i - 最後 番目(j) + 8 番目(c) を 1 で置を換える
queen i + 1
最後 番目(j) 番目(a) を c で置を換える
i + 最後 番目(j) - 1 番目(b) を c で置を換える
i - 最後 番目(j) + 8 番目(c) を c で置を換える
でなければ
printQueen
最後 番目(j) を 最後 番目(j) + 1 で置を換える
最後 番目(j) から削除する
```

## 変数(配列)の意味

- $x[i]$  :  $i$  行目に置く王妃の列位置
- $a[j]$  :  $j$  列目に王妃が置かれているか
- $b[i + j - 1]$  : 右上に向かう対角線上( $i + j - 1$ )に王妃が置かれているか
- $c[i - j + 8]$  : 右下に向かう対角線上( $i - j + 8$ )に王妃が置かれているか

# バックトラック(5)

- [地球にやさしいアルゴリズム>第5回>隣の区画と異なる色で地図を四色に塗り分ける\(4色問題\)](#)

## ☆ 4色問題

どんな地図でも4色あれば隣接したエリアとは別の色で塗ることができる、というのが4色問題です。4色問題の証明は長い間、数学(グラフ理論)の難問の1つだったそうです。

Scratchで試してみよう！

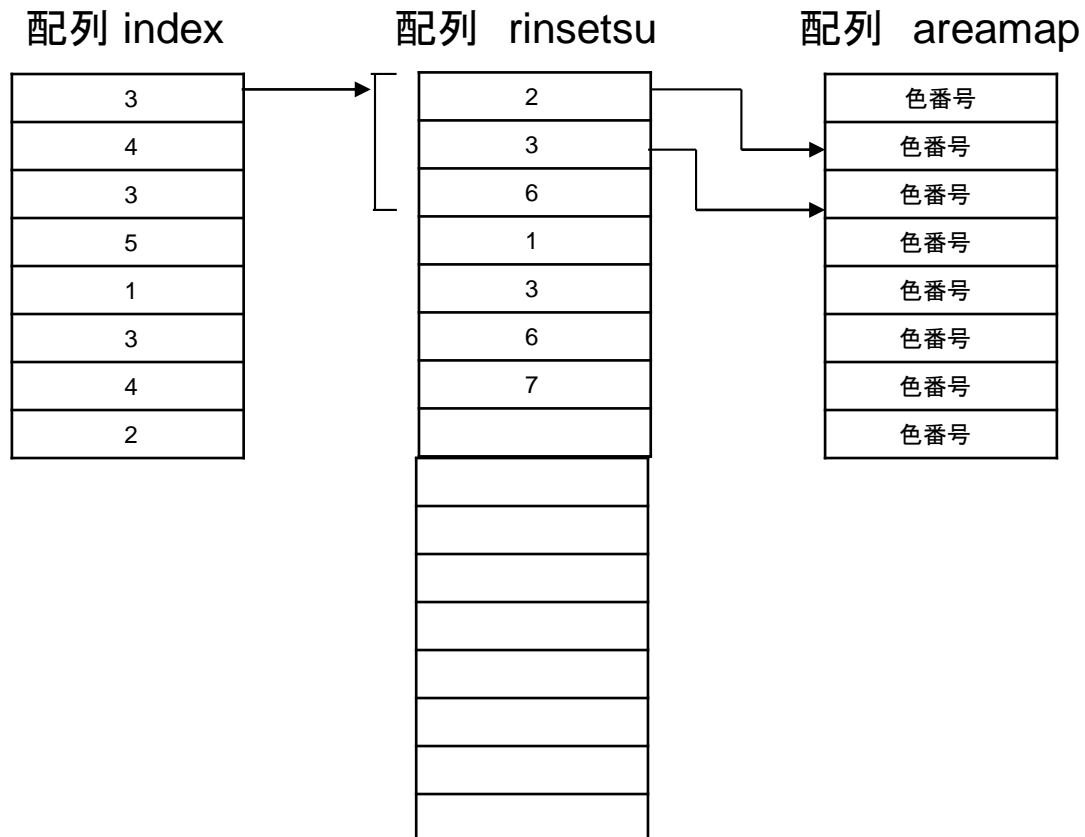
<https://scratch.mit.edu/projects/89672071/>

上記のプロジェクトのネコのスクリプトの「スペースキー」が押されたときの処理(4色問題の解答を表示する処理)には右図の処理が8回記述されている。データ構造を工夫して、スクリプトがもっと短くなるようにリミックスしてみよう！



# バックトラック(6)

- [地球にやさしいアルゴリズム>第5回>隣の区画と異なる色で地図を四色に塗り分ける\(4色問題\)](#)



# バックトラック(7)

[ 解答例 ]

8王妃(繰り返し版)

<https://scratch.mit.edu/projects/153686514/>

4色問題(データ構造改良版)

<https://scratch.mit.edu/projects/153691580/>