

付録 「学校に行く道順」のプログラム

アンプラグドで考えた「学校に行く道順」を検証するためのプログラムの使い方を説明します。

●プログラムの読み込み

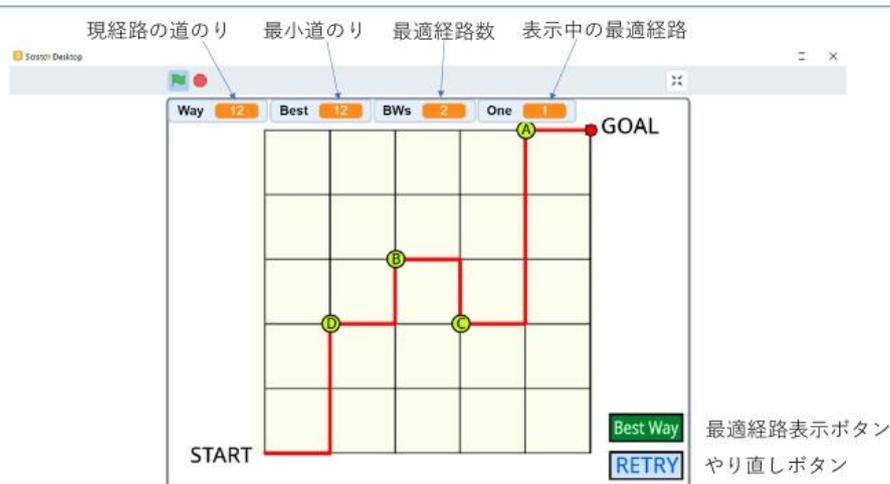
Scratch デスクトップの「ファイル」から、次のファイルを読み込みます。

2-x_route.sb3

●全画面表示

ステージを全画面にします。

家から学校まで行く道順



Copyright © ASAI Noboru, 2020

1

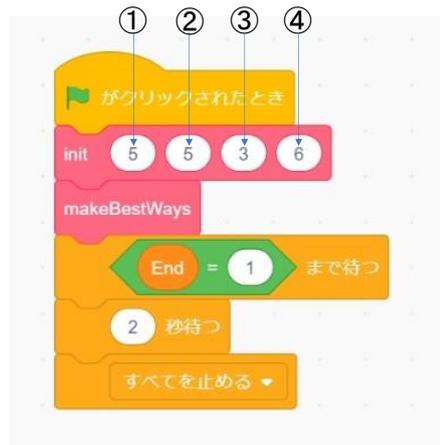
●使い方

- ・[実行]ボタンを左クリックすると、マス目の交点に友人 6 人まで A~F で表示されます。
- ・「Best」に最小の道のりが表示されます。
- ・「BW's」に最適経路(最小の道のりで行けるルート)の数が表示されます。
- ・A~F キーを押すと、現在の位置からそこまで移動します。
- ・移動のたびにそこまでの道のりが「Way」に表示されます。
- ・G キーで GOAL まで移動し、実行終了です。Way=Best なら最適経路です。
- ・G キーを押したときに、まだ通過していない友人があれば、文字を白抜き表示します。
- ・[RETRY]ボタンを押すと、同じ友人配置で、START からやり直します。
- ・[Best Way]ボタンを押すと、最適経路を表示します。
- ・「One」に表示中の最適経路の番号が表示されます。
- ・最適経路が複数あるときは、[Best Way]ボタンを押すたびに、次の最適経路を表示します。

●初期設定の変更方法

ステージの「がクリックされたとき」の init 定義ブロック呼び出しのパラメータで、「マス目の数」と「友人の数」を変更できます。

家から学校まで行く道順（初期設定）



- ①横のマス数
- ②縦のマス数
- ③友人の最少人数
- ④友人の最多人数

- ① 横のマス数を1以上の整数で指定します。
- ② 縦のマス数を1以上の整数で指定します。
- ③ 友人の最少人数を指定します。0以上、最多人数以下の整数を指定します。
- ④ 友人の最多人数を指定します。最少人数以上、6以下の整数を指定します。

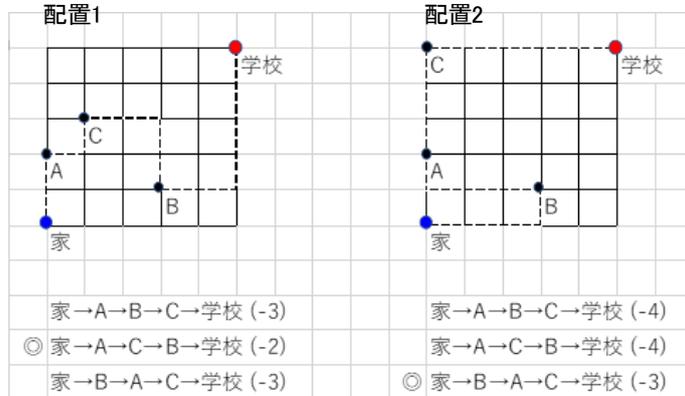
初期設定のパラメータは、不当な場合は自動的に補正されますが、あまり大きい数は指定しないでください。友人の人数は6以下に補正されますが、マス目はそのまま使いますので、大きい数を指定すると、コンピュータによってはメモリ不足などで、動きがおかしくなることがあります。本来はそのような場合のエラー処理を考慮すべきですが、このプログラムではそこまで考慮していないので、ご了承ください

●プログラムの解説

「学校に行く最適な道順」という問題は、組み合わせ最適化問題というとても面倒な問題です。可能な経路の道のりをすべて数えて比較し、一番小さい値の経路を選ぶ、という手順になります。友人の数が少ないときは簡単に解けますが、人数が増えるにつれて比較の回数がどんどん多くなり、コンピュータを使っても大変なのです。友人の数を n とすると、比較回数の計算量は $n!$ (階乗: n から1までのすべての数の掛け算)になります。これは 2^n よりも多い計算量です。

友人が2人の場合は、後戻りの数の比較は両者の位置関係ですぐわかりましたが、3人以上になると簡単にはわからないので、後戻りの数を計算して全経路を比較する必要があります。

家から学校まで行く道順（友人3人以上）



後戻り数を比較する
組み合わせ最適化問題
友人n人の場合、n!通り

()内は後戻りの数、◎が最適経路

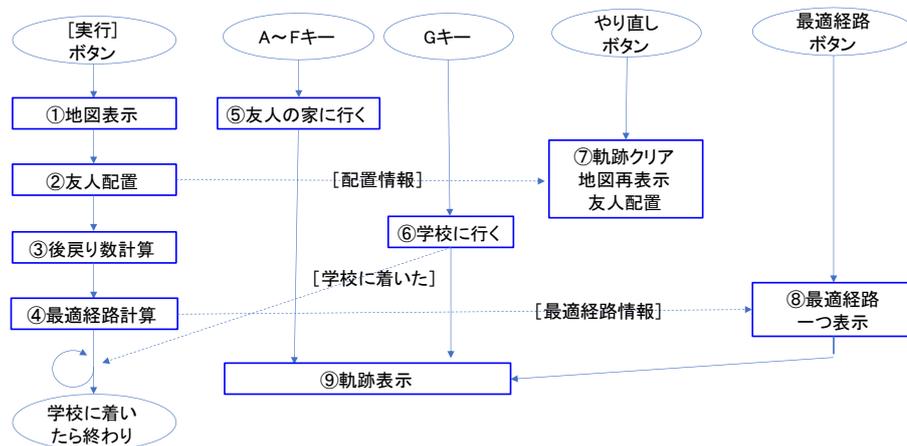
道のり=横マス数+縦マス数+2×後戻り数

Copyright © ASAI Noboru, 2020

3

それでも、「すべての経路の道のりを比較する」というやり方よりは、「後戻りの数を見て不要な経路を捨てる」という工夫をすれば、断然良いアルゴリズムになります。今回のプログラムも、そのようなアルゴリズムに基づいてつくりました。詳細は述べませんが、大きな流れを下図に示します。

家から学校まで行く道順（友人3人以上の大きな流れ）



Copyright © ASAI Noboru, 2020

4

「④最適経路計算」では、「繰り返し」と「呼び出し」を合わせたような「再帰」という技法を使っています。「再帰」は関数の中で自分自身を呼び出す、という、いわば鏡に鏡を映すような呼び出し方です。関数の頭から繰り返すのですが、Scratchではブロックに局所的な変数をつくれないので、本書では解説しなかったリストを使って実現しています。興味のある方は参考にしてください。