

SuperCon2018 認定問題と予選問題

1 自己回避ウォークと自己回避ループ

平面上に描かれた格子を考えよう。格子の隣り合う交点 N 個を次々につないでいくと、ひとつの「道」が作られる。ここで、道は交差できないという条件を課す。つまり、道は同じ交点を二度通ることにはないとする。このような道を**自己回避ウォーク** (Self-Avoiding Walk、略して SAW) と呼ぶ。SAW はたとえば高分子 (ポリエチレンのような人工高分子から DNA やタンパク質などの生体高分子まで) の簡単なモデルとしてよく用いられ、今も盛んに研究されている。また、最後に出発点に戻る SAW を**自己回避ループ** (Self-Avoiding Loop、略して SAL) と呼ぶ。これはたとえば環状 DNA (人間のミトコンドリアが持つ DNA は環状である) のモデルと考えることができるだろう。図 1 は SAW、図 2 は SAL の例である (通る交点の数はどちらも 20 だが、道の長さはひとつ違うことに注意)。

これら、自己回避ウォークや自己回避ループの総数を数えようというのが今回の問題である。

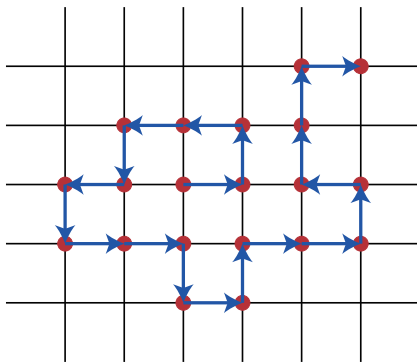


図 1 長さ 19 の SAW の例

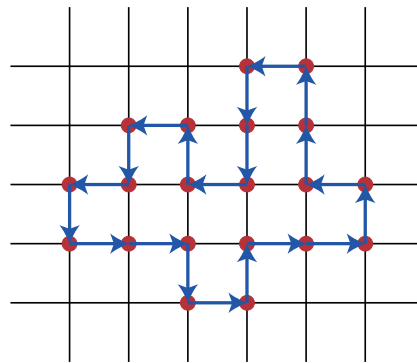


図 2 長さ 20 の SAL の例

2 問題

2.1 SuperCon 3 級認定問題

問題として与えられた長さ 18 の道 (つまりスタートを含めて 19 個の交点をつなぐ) が SAW かどうかを判定するプログラムを作れ。

2.1.1 3 級認定問題詳細

進行方向に図 3 のように数字を割り当て、道は進行方向の列として表現される。たとえば、図 4 の長さ 11 の道は 0 0 1 2 1 0 0 3 3 3 0 と表される (この例は SAW である)。このような進行方向

の列が問題として与えられるので、それで表現されている道が SAW かどうか (つまり、同じ交点を二度以上通っていないかどうか) を判定し、SAW であれば 1 を SAW でなければ 0 を出力するプログラムを作成する。

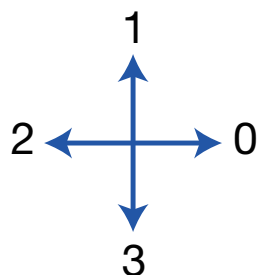


図3 進行方向に対する数値の割り当て

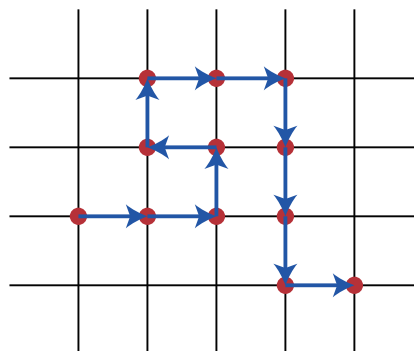


図4 0 0 1 2 1 0 0 3 3 3 0 と表される道

ヘッダーファイル `sc3.h`(詳細は後述) に定義された `scInput` 関数を実行すると、以下の**グローバル配列**に問題が格納される。

- `scD[18]`: 道の進行方向を表す数値 (0,1,2,3 のいずれか) の列をひとつずつ格納。
`scD[0]` が最初の進行方向で、以下、配列の順に `scD[17]` までで長さ 18 の道が指定される。

2.2 SuperCon 2 級認定問題

格子上的特定の交点 (原点と表記する) からスタートする長さ 12(つまりスタートを含めて 13 個の交点をつなぐ) の SAW のうち、問題で指定される特定の点 (原点以外の一点) を通るものの総数を求めるプログラムを作れ。

2.2.1 2 級認定問題詳細

交点に図 5 のように座標を割り当てる (格子が図の範囲に限られるという意味ではない。図の範囲より外側にもいくらでも広がっている)。原点は (0,0) である。入力として原点以外の 1 点の座標が与えられる。原点からスタートして指定された点を通る長さ 12 の SAW の総数を出力するプログラムを作成する。図 6 は (0,0) をスタートして途中で交点 (2,1) を通る SAW の例である。なお、指定された交点を道の途中ではなく終点とする SAW も数えるものとする。また、指定される交点を通る SAW は最低でもひとつは必ずある。

ヘッダーファイル `sc2.h`(詳細は後述) に定義された `scInput` 関数を実行すると、以下の**グローバル配列**に問題が格納される。

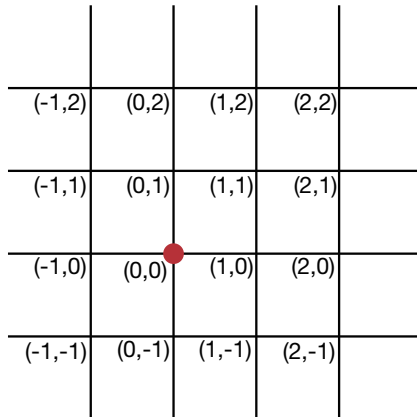


図5 交点に対する座標の割り当て

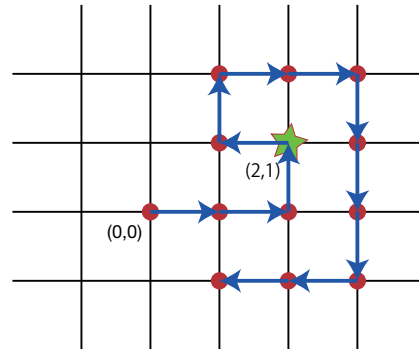


図6 原点から出発して途中で(2,1)を通る長さ12のSAWの例

- `scP[2]`: 道を通るべき点の横座標が `scP[0]` に、縦座標が `scP[1]` に格納される。

2.3 SuperCon 1 級認定兼予選問題

格子上に障害物が置かれていて通れない交点がある。このとき、格子の原点を通る長さ N の SAL の総数を数えるプログラムを作れ。 $14 \leq N \leq 20$ (必ず偶数である) とする。

2.3.1 1 級認定兼予選問題詳細

2 級問題と同様に交点に座標を割り当てる (図 5)。障害物が置かれる交点の数 M は 10 個以内とし ($1 \leq M \leq 10$)、その数 M とそれぞれの座標が入力として与えられる。また、SAL の長さ N も入力として与えられる ($14 \leq N \leq 20$ の偶数)。このとき、 $(0, 0)$ を出発して障害物の置かれた交点を通らずに原点に戻ってくる長さ N の SAL の総数を求めるプログラムを作る。障害物は少なくともひとつは SAL が作れるように配置される。なお、同じ交点の組を逆順に巡る SAL は別々のものとして (つまり、ふたつとして) 数える。また、平行移動で重なる同じ形の SAL もすべて別々のものとして数える。

ヘッダーファイル `sc1.h` (詳細は後述) に定義された `scInput` 関数を実行すると、以下の **グローバル変数・グローバル配列** に問題が格納される。

- `scN`: SAL の長さ N 。14,16,18,20 のいずれかである。
- `scM`: 障害物の数 M 。1 以上 10 以下の整数である。
- `scB[2][10]`: 障害物の位置を表す座標を格納する配列である。`scB[0][n]` は $n + 1$ 番目の障害物の横座標、また `scB[1][n]` は縦座標を表す。 $n > M - 1$ の配列の要素は意味を持たない。

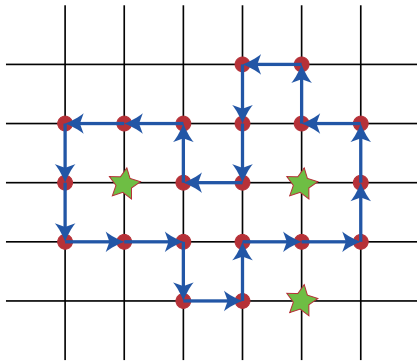


図7 三つの障害物を避けた SAL の例。星が障害物を表す

3 級の認定基準と予選の実考基準

各級とも、5問を解き、すべてに正答すれば級を認定する。それぞれ、1問を解くプログラムを提出してもらい、運営側で問題を変えて5回実行する。制限時間として、5問の合計実行時間合計の上限を2,3級は5分、1級は10分とし、それ以内にすべて正答できた場合のみ級を認定する。計算実行時間にはデータの入出力時間は含まれない(下で説明する入出力関数に時間測定が含まれており、入力終了時から出力直前までの時間を計測する)。

1級の問題は SuperCon 予選問題を兼ねているので、本選参加希望チームは1級問題の解答プログラムを提出する(本選参加を希望せず、1級認定のみを希望する場合は応募時にその旨を書くこと)。東西それぞれ、正答数の多いものから順に10チームを予選通過とする。正答数が同じ場合、合計の計算実行時間が短いチームから順に選ぶ。ただし、級認定と同様に合計実行時間の上限を10分とし、制限時間内に解けた問題数のみを数える。5問で実行時間に差がつかない場合には問題数や制限時間を増やすこともありうる。なお、制限時間内に全問正解せずに予選通過となった場合、本選には出場できるが1級は認定されない(運営側としては、予選通過チームは全問正解するはずと期待している)。

4 ヘッダーファイルと入出力関数

各級ごとにヘッダーファイルを用意してあるので、それをインクルードする。ヘッダーファイル名は3,2,1級それぞれ、sc3.h、sc2.h、sc1.hである。これをプログラムと同じディレクトリー(フォルダー)に置いて

```
#include "sc1.h"
```

などとする。たとえば、stdin.h と stdlib.h を使う場合、プログラムの冒頭部は

```
#include<stdio.h>
#include<stdlib.h>
#include "sc1.h"
```

のようになる。なお、C++ 言語を使う場合にも `iostream` ではなく `stdio.h` をインクルードすること。

ヘッダーファイル内では問題詳細で述べたグローバル変数・グローバル配列のほかに、以下のように入出力関数と時間測定のための変数が定義されている。

- `void scInput()`: 入力関数。標準入力から問題データを読み、グローバル変数・グローバル配列に格納する。また、入力後に時刻 `scStartTime` を設定する
- `void scOutput(int s)`: 出力関数。答えの数値 `s` を標準出力に出力する。また、出力前に時刻 `scEndTime` を設定し、答えに続いて計算時間として `scEndTime-scStartTime` を出力する。
- `scStartTime`, `scEndTime`: 時刻を格納する変数。

問題を読み込むには `scInput` 関数を引数なしで実行する。`scInput` 関数は必ずプログラムの最初の実行文であること。`scInput` の前に実行文を書いてはならない。解答の数値は `scOutput` 関数に渡して実行することで出力される。

なお、ヘッダーファイルは改変してはならない。また、入出力は必ず `scInput` と `scOutput` のみで行ない、他の出力をプログラムに含めてはならない。また、`scStartTime` と `scEndTime` の値はプログラム中で変えてはならない。これらの注意に反するプログラムは失格とする。

4.1 言語と実行環境その他の注意

プログラムは ANSI 準拠の C 言語 (C99) または C++ 言語 (C++03) で書く。また、プログラムは分割せずひとつのファイルであること (ヘッダーファイルは提出不要)。独自のヘッダーファイルやライブラリーは使用できないので、プログラム開発に統合開発環境を使用する参加者はくれぐれも注意すること。

審査は 2.5GHz の Intel Xeon CPU を搭載した LINUX マシン上で実行する。コンパイラは gcc バージョン 4.4.7 を使い、プログラムが C 言語で書かれている場合は

```
gcc -O -lm -std=c99 プログラム名. c
```

C++ 言語で書かれている場合は

```
g++ -O -lm -std=c++03 プログラム名.cpp
```

とコンパイルする

4.2 サンプル問題

問題データのサンプルをそれぞれひとつウェブサイトに掲載する。さらにテストデータが必要なときはこれらを参考に独自に作ってもらいたい。

5 提出方法

ひとつの級にだけ応募してもよいし、複数の級に応募してもよい。作成したプログラムはファイル名を 1,2,3 級それぞれ、`sc1.c`、`sc2.c`、`sc3.c` とし (C++ の場合は拡張子を `cpp` とする。たとえば `sc1.cpp`)、メールの添付ファイルとして提出する。メールには必ず学校名、全員の名前と学年、連絡先メールアドレスを明記すること。級認定のみの応募はひとりでも構わないが、SuperCon 予選に応募する場合には 2 ないし 3 名のチームを作り、チーム名も明記すること (予選通過後のチーム構成変更は認められないので注意)。1 級応募者は何も書かれていなければ予選参加とみなすので、級認定のみ希望の場合はその旨を明記すること。

提出および問い合わせのメールアドレスは SuperCon ウェブサイトを参照のこと。