

スーパーコン23予選・認定問題: 地形推定問題

1 地形推定問題

でこぼこの地形の上を進んでゆく M 人のハイカー (ハイキングをする人) $m = 0, 1, 2, \dots, M - 1$ がいる. 図1のような格子があり, 各格子点 (x, y) での高さを $V(x, y)$ とする. ここで y は $0, 1, 2, \dots, L - 1$ の値を取り, x は $0, 1, 2, \dots, N - 1$ の値を取る. ハイカーは, 時刻 $t = 0$ では正方格子の下端 $y = 0$ のどこかにいる. m 番目のハイカーに注目し, その x 座標を $x_m(0)$ としよう. またこの時ハイカーは「エネルギー」 $E_m(0)$ を持っている. ここを出発して時刻 $t = 1, 2, \dots$ では $y = 1, 2, \dots$ のように一步一步 y 軸の正の方向に進んでゆく. 今, ある時刻 t でのこのハイカーの x 座標を $x_m(t)$, エネルギーを $E_m(t)$ としよう. 次の時刻の x 座標 $x_m(t+1)$ として許されるのは, $x_m(t) + 1, x_m(t), x_m(t) - 1$ のいずれかだけである. 高さの変化分は $\Delta V = V(x_m(t+1), y+1) - V(x_m(t), y)$ となる. この時, ハイカーのエネルギーは変化し, 新しいエネルギーは $E_m(t+1) = E_m(t) - \Delta V$ となる. ΔV が正ならエネルギーを消費し, 負ならばエネルギーを稼ぐということである. ハイカーはこれが負にならない範囲で, なるべく高さを稼ごうとする. つまり $E_m(t+1)$ が負にならない範囲で, ΔV が最大になる方向に進む. これがハイカーの進み方である. 3つの方向のどちらにいても $E_m(t+1)$ が負になってしまう場合は, そこでこのハイカーは行き止まりとする. 行き止まらずに $y = L - 1$ にたどり着けばゴールである.

でこぼこ地形を表す $V(x, y)$ は, どの y においても $0, 1, 2, \dots, N - 1$ のいずれかの値をとり, かつ同じ値となる x はないとする. つまりどの y においても $V(0, y), V(1, y), \dots, V(N - 1, y)$ は $0, 1, 2, \dots, N - 1$ を適当に並べ替えたものとする.

全ての点での V の値, $V(x, y)$ が与えられ, 個々のハイカーの最初の x 座標, $x_m(0)$ が与えられれば, それぞれゴールするかしらないか, ゴールする場合はどこに着くかを求めることができる (3, 2級認定問題). 逆に, 次のような問題も考えることができる. ハイカーの最初の x 座標, $x_m(0)$ がすべてのハイカー $m = 0, 1, 2, \dots, M - 1$ について与えられる. また, 全員がゴールするとして, そのときの x 座標 $x_m(L - 1)$ も与えられる. このようなことが実現する地形 $V(x, y)$ を推定することを考えよう (1級認定問題兼予選問題)

境界条件

各時刻にてハイカーのとりうる x 座標は, $0, 1, 2, \dots, N - 1$ である. したがって $x = 0$ にいるとき, 次の時刻で許されるのは $x = 0, 1$ のみで $x = -1$ は排除する. 同様に $x = N - 1$ にいるとき, 次の時刻で許されるのは $x = N - 2, N - 1$ のみで $x = N$ は排除する.

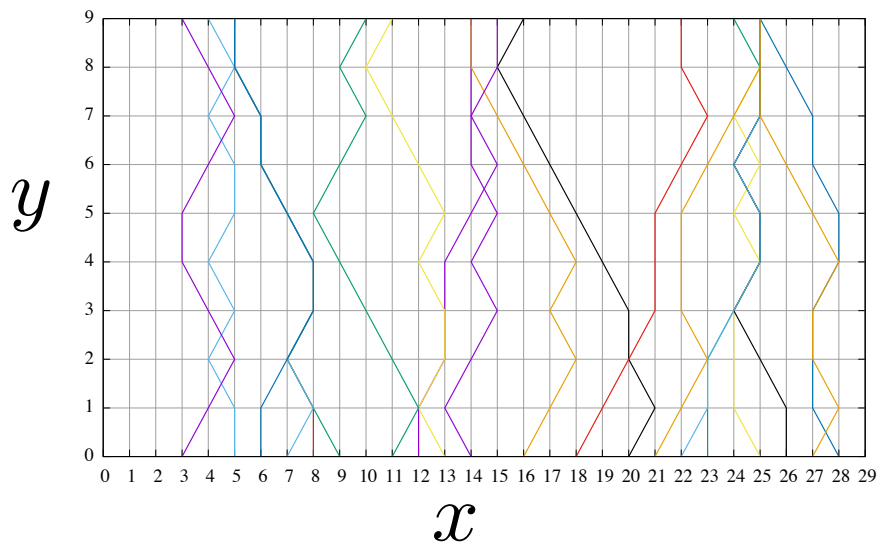


図 1 ハイカーの軌跡 $x_m(t)$ の例. この例では時刻 $t = 0$ では全てのハイカーが $E_0 = 4$ のエネルギーを持っている.

2 問題

2.1 SuperCon 3 級認定問題

地形 $V[I][J]$ ($I = 0, 1, 2, \dots, N - 1, J = 0, 1, 2, \dots, L - 1$) と時刻 $t = 0$ でのハイカーの x 座標 $X_s[m]$ ($m = 0, 1, 2, \dots, M - 1$) が問題として与えられる。ゴールにおけるハイカーの x 座標 $X_g[m]$ ($m = 0, 1, 2, \dots, M - 1$) を求めるプログラムを作れ。ただしこの問題では簡単のためにエネルギー E による制限はないとする。つまり各ステップで、ハイカーは ΔV が最大となる方向に進む。

入力

標準入力から以下の形式の入力ファイルを読み込む。このための関数はあとで説明するように提供するヘッダーファイルに含まれる。

```
V[0][0]
V[1][0]
⋮
V[N - 1][L - 1]
Xs[0]
Xs[1]
⋮
Xs[M - 1]
```

出力

標準出力に以下の形式のファイルを出力する。このための関数もあとで説明するように提供されるヘッダーファイルに含まれる。

```
Xg[0]
Xg[1]
⋮
Xg[M - 1]
入力読み込み終了からの経過時間 (秒)
```

2.1.1 3 級認定問題詳細

ヘッダーファイル `sc3.h` に必要な関数と配列が定義されているので,

```
#include "sc3.h"
```

として使うこと. 定義されているものは以下である.

```
void SC_input() : 問題を標準入力から読み込む関数
void SC_output() : 解答を標準出力に出力する関数
# define SC_L 10 : 正方格子の y 軸方向の長さ
# define SC_N 30 : 正方格子の x 軸方向の長さ
# define SC_M 10: ハイカーの数
int SC_V[SC_N][SC_L] : 問題の地形を格納する 2 次元整数配列
int SC_Xs[SC_M] : 問題のハイカーの最初の位置を格納する 1 次元整数配列
int SC_ans[SC_M] : 解答を格納する 1 次元整数配列
```

2.1.2 3 級認定基準

1 問を解くプログラムを作成して応募する. 問題は 10 問与えられ, 問題を変えてプログラムを 10 回実行する. 制限時間内にすべてに正解した場合に 3 級を認定する. 制限時間は入出力を除く計算時間で 10 問合計 1 分とする.

2.2 SuperCon 2 級認定問題

地形 $V[I][J]$ ($I = 0, 1, 2, \dots, N - 1, J = 0, 1, 2, \dots, L - 1$) と時刻 $t = 0$ でのハイカーの x 座標 $X_s[m]$ ($m = 0, 1, 2, \dots, M - 1$) が問題として与えられる. ゴールにおけるハイカーの x 座標 $X_g[m]$ ($m = 0, 1, 2, \dots, M - 1$) を求めるプログラムを作れ. なお, ゴールできず行き止まりとなるハイカーの場合は $X_g[m] = -1$ とせよ. 時刻 $t = 0$ で全てのハイカーは同じエネルギー E_0 をもつ. その値はヘッダーファイルで指定されるものを用いよ.

入力

標準入力から以下の形式の入力ファイルを読み込む. このための関数はあとで説明するように提供するヘッダーファイルに含まれる.

```
V[0][0]
V[1][0]
⋮
V[N - 1][L - 1]
Xs[0]
Xs[1]
⋮
Xs[M - 1]
```

出力

標準出力に以下の形式のファイルを出力する. このための関数もあとで説明するように提供されるヘッダーファイルに含まれる.

```
Xg[0]
Xg[1]
⋮
Xg[M - 1]
入力読み込み終了からの経過時間 (秒)
```

2.2.1 2 級認定問題詳細

ヘッダーファイル `sc2.h` に必要な関数と配列が定義されているので,

```
#include "sc2.h"
```

として使うこと. 定義されているものは以下である.

```
void SC_input() : 問題を標準入力から読み込む関数
void SC_output() : 解答を標準出力に出力する関数
# define SC_L 10 : 正方格子の y 軸方向の長さ
# define SC_N 30 : 正方格子の x 軸方向の長さ
# define SC_M 20: ハイカーの数
# define SC_E0 4: 時刻 t=0 でのハイカーのエネルギー
int SC_V[SC_N][SC_L] : 問題の地形を格納する 2 次元整数配列
int SC_Xs[SC_M] : 問題のハイカーの最初の位置を格納する 1 次元整数配列
int SC_ans[SC_M] : 解答を格納する 1 次元整数配列
```

2.2.2 2 級認定基準

1 問を解くプログラムを作成して応募する. 問題は 10 問与えられ, 問題を変えてプログラムを 10 回実行する. 制限時間内にすべてに正解した場合に 2 級を認定する. 制限時間は入出力を除く計算時間で 10 問合計 1 分とする.

2.3 SuperCon 1 級認定問題兼予選問題

時刻 $t = 0$ でのハイカーの x 座標 $X_s[m]$ ($m = 0, 1, 2, \dots, M - 1$) と時刻 $t = L - 1$ でのハイカーの x 座標 $X_g[m]$ ($m = 0, 1, 2, \dots, M - 1$) が問題として与えられる。これが実現するための地形 $V[I][J]$ ($I = 0, 1, 2, \dots, N - 1, J = 0, 1, 2, \dots, L - 1$) を一つ求めるプログラムを作成せよ。時刻 $t = 0$ で全てのハイカーは同じエネルギー E_0 をもつ。その値はヘッダーファイルで指定されるものを用いよ。なお、全てのハイカーがゴールできる場合のみが問題として与えられる。また、正解はひとつとは限らないが、どれか一つの正解を求めればよい。

入力

標準入力から以下の形式の入力ファイルを読み込む。このための関数はあとで説明するように提供するヘッダーファイルに含まれる。

```
X_s[0]
X_s[1]
⋮
X_s[M - 1]
X_g[0]
X_g[1]
⋮
X_g[M - 1]
```

出力

標準出力に以下の形式のファイルを出力する。このための関数もあとで説明するように提供されるヘッダーファイルに含まれる。

```
V[0][0]
V[1][0]
⋮
V[N - 1][L - 1]
入力読み込み終了からの経過時間 (秒)
```

地形に対する条件

どの y においても $V(0, y), V(1, y), \dots, V(N - 1, y)$ は $0, 1, 2, \dots, N - 1$ を適当に並べ替えたものとする。これを満たさないものは不適である。

2.3.1 1 級認定問題詳細

ヘッダーファイル `sc1.h` に必要な関数と配列が定義されているので、

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

として使うこと。定義されているものは以下である。

```
void SC_input() : 問題を標準入力から読み込む関数
void SC_output() : 解答を標準出力に出力する関数
# define SC_L 10 : 正方格子の y 軸方向の長さ
# define SC_N 30 : 正方格子の x 軸方向の長さ
# define SC_M 20: ハイカーの数
# define SC_E0 4: 時刻 t=0 でのハイカーのエネルギー
int SC_Xs[SC_M] : 問題のハイカーの最初の位置を格納する 1 次元整数配列
int SC_Xg[SC_M] : 問題のハイカーの最後の位置を格納する 1 次元整数配列
int SC_ans[SC_N][SC_L] : 解答を格納する 2 次元整数配列
```

2.3.2 1 級認定基準

1 問を解くプログラムを作成して応募する。問題は 10 問与えられ、問題を変えてプログラムを 10 回実行する。制限時間内にすべての問題について結果が出力され、すべて正解である場合に 1 級を認定する。制限時間は入出力を除いて 10 問合計 1 分とする。

2.3.3 本選出場チーム選考基準

問題は 100 問与えられる。制限時間内に結果が出力された問題について正解の数の合計でランキングし、上位 20 チームを本戦出場チームとする。制限時間は入出力を除いて 100 問合計 1 分とする。以上の基準でもし 20 チーム決まらなければ、問題数を増やして（制限時間もそれに比例して増やす）順位を決定する。また正解の数で差がないチーム同士は、計算時間の合計でランキングする。成績がほとんど変わらない時には、レポートを参考に、アルゴリズム上の工夫が見られたチームを優先する。なお、本選出場と 1 級認定の基準は違うので、本選出場チームでも 1 級認定が与えられない場合がある。

3 言語と実行環境その他の注意

- 言語: C++ を使用すること。
- OS: 審査は macOS Ventura 13.3.1(a), CPU: プロセッサ Apple M1, メモリ 16GB を搭載したコンピュータ上でおこなう。
- コンパイラは Homebrew GCC 14.0.3 を使用し、以下のように -O2 オプションでコンパイルする。

```
g++-13 -O2 sc3.cpp
```
- 出力に関する注意: 実行時に問題は標準入力から読まれ、解答は標準出力に出力される。ヘッダーファイルに定義された入力、出力用の関数を用いること。すべての級で、SC_input 関数は必ずプログラムの最初の実行文であること。つまり、SC_input の前に実行文を書いてはならない。入出力は SC_input と SC_output 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
- 計算時間は time.h に含まれる clock() 関数で測定する。入力終了後に測定を開始し、出力直前までの所要時間を測る。また、開始時刻と終了時刻を表す変数 SC_starttime と SC_endtime の値はプログラム中で参照してよいが、変えてはならない。なお、ヘッダーファイルは改変してはならない。これらの注意に反するプログラムは失格とする。

4 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 級は `sc1.cpp`

2 級は `sc2.cpp`

3 級は `sc3.cpp`

として、以下のファイルと一緒に zip ファイルにまとめて Google フォーム「SuperCon2023 予選・級認定申込」(<https://forms.gle/vr1FLMEsCgzsKxVRA>) から提出する。zip ファイルの名前は” <チーム名> .zip”にすること。整理の都合上、予選に応募しない場合も (ひとりでも) チーム名はつけること。なお、チーム名は 8 文字以内の英数字とします。ただし、1 文字目は英字でなければなりません。

zip ファイルにまとめて提出するものは以下の 3 つである。

- 提出プログラム: `sc1.cpp`, `sc2.cpp`, `sc3.cpp` のうち応募する級に対応するもの。予選応募には `sc1.cpp` が必須である。
- プログラム実行環境の説明 (下で説明するサンプル問題の一つに対する実行結果と計算時間, 使用コンパイラなど) ” <チーム名> -env.txt”
- アルゴリズムのポイントや工夫などをまとめたレポート: ファイル名はテキストファイルの場合は” <チーム名> -algo.txt”, word ファイルの場合は” <チーム名> -algo.docx”, pdf ファイルの場合は” <チーム名> -algo.pdf”

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上記 Google フォームで明記すること。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。

5 注意事項

問題解法を他のチーム (学校が同じか違うかにかかわらず) と相談してはならない。実施委員会としてはこのことを監督する手段はないが, 厳守をお願いする。また, 募集締め切りまでは, 問題解法に関して SNS やブログ等, チームメンバー以外の目に触れる場所に投

稿することを禁止する。解法に関する内容を含まない「SuperCon 予選問題の解きかたを議論しています」などの投稿は自由である。募集締め切り後には解法についても SNS やブログに投稿して構わない。

6 問題のサンプル

問題のサンプルをウェブサイトにて公開する。