

# スーパーコン 2 4 予選・認定問題：格子状の街の移動

2024 年 6 月 5 日

## 1 問題説明

SC 街は起伏が激しい格子状（碁盤の目状）の街である。SC 社は SC 街の道案内アプリを作成しており、ユーザが SC 街の 2 つの地点を入力したら 2 地点間の最短の移動時間を計算するというサービスを提供している。しかし、SC 社のプログラムは、（同じ地点を 2 度通らない経路を全て列挙してそのうちの移動時間が最短のものを求めるという方針のプログラムで）非常に効率が悪く、現実的な時間では計算結果が返ってこないことで悪名高いアプリであった。あなたの仕事は、このプログラムを改善することである。道案内アプリを通してユーザから（2 地点の情報に関する）クエリがたくさん届くので、できるだけ高速に処理しよう。

より具体的な問題設定は、次の通りである。まず  $N \times N$  の 2 次元格子が与えられる。左上の頂点を  $(0, 0)$  とし、右方向に  $x$  回、下に  $y$  回進んだ先の頂点を  $(x, y)$  とする。隣接する（頂点  $(x, y)$  に隣接する頂点とは、 $(x - 1, y)$ 、 $(x + 1, y)$ 、 $(x, y - 1)$ 、 $(x, y + 1)$  の 4 つの頂点をさす）頂点間には移動にかかる時間が設定されており、頂点  $(x, y)$  と頂点  $(x + 1, y)$  の間の移動にかかる時間は  $h_{x,y}$  ( $0 \leq x \leq N - 2$  かつ  $0 \leq y \leq N - 1$ )、頂点  $(x, y)$  と頂点  $(x, y + 1)$  の間の移動にかかる時間は  $v_{x,y}$  ( $0 \leq x \leq N - 1$  かつ  $0 \leq y \leq N - 2$ ) である。なお、頂点間の移動にかかる時間は向きによらない。図 1 に問題例を示す。

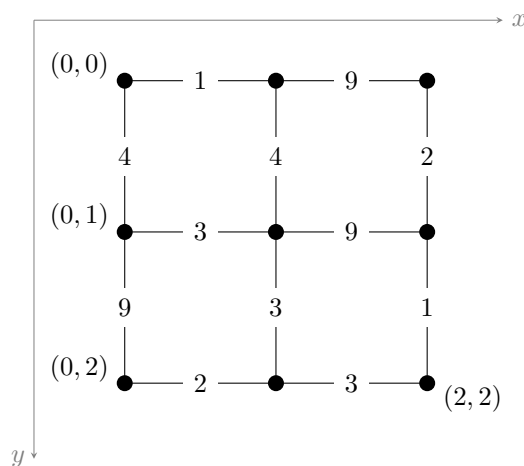
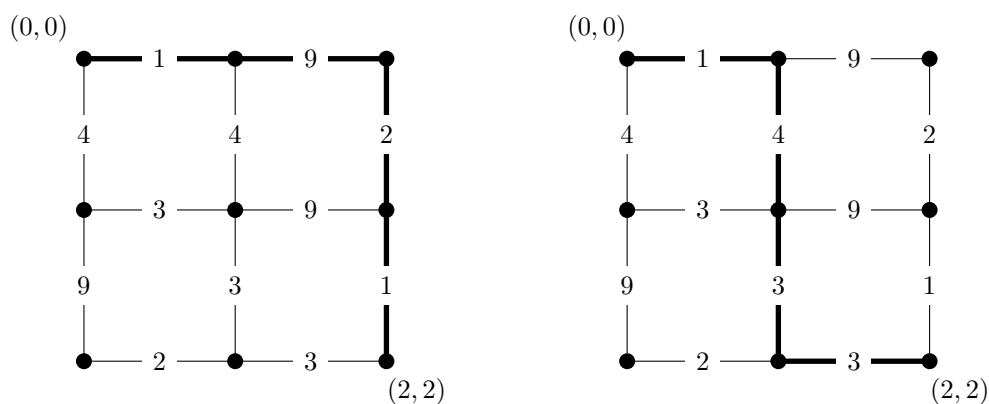


図 1: 問題例 1 ( $N = 3$ )

本問では、頂点  $(a, b)$  から頂点  $(c, d)$  の移動にかかる最短の時間（頂点  $(a, b)$  からスタートして隣接する頂点への移動を繰り返して頂点  $(c, d)$  に移動するまでにかかる時間の合計の最小値） $r$  を考えたい。たとえば、図 1 において、頂点  $(0, 0)$  から頂点  $(2, 2)$  に図 2a の太線で示された経路をたどって移動した時の時間は 13 で

ある。しかし、図 2b の太線で示された経路では移動時間が 11 でより短く、この時が最小である。したがっ



(a) 移動例 1 (この時の移動時間は 13) (b) 移動例 2 (この時の移動時間は 11 でこれが最小)

図 2: 問題例 1 における (0,0) から (2,2) への移動の例

て、図 1 において、 $(a,b) = (0,0)$  かつ  $(c,d) = (2,2)$  の時、 $r = 11$  である。本問では、このようなクエリがひとつの問題例につき  $Q$  個与えられる。 $Q$  個のクエリのうち  $i$  ( $1 \leq i \leq Q$ ) 番目のクエリには頂点  $(a_i, b_i)$  と頂点  $(c_i, d_i)$  の情報が与えられるので、頂点  $(a_i, b_i)$  から頂点  $(c_i, d_i)$  への移動にかかる最短の時間を求めてほしい。この処理をできるだけ高速におこなうプログラムを作成するのがあなたの仕事である。

## 入力

入力は以下の形式で与えられる：

```

N
h0,0 h1,0 ... hN-2,0
v0,0 v1,0 ... vN-1,0
h0,1 h1,1 ... hN-2,1
⋮
v0,N-2 v1,N-2 ... vN-1,N-2
h0,N-1 h1,N-1 ... hN-2,N-1
Q
a1 b1 c1 d1
⋮
aQ bQ cQ dQ

```

- 1 行目には整数  $N$  が与えられる。
- 続く  $2N - 1$  行のうち、
  - $2y + 1$  ( $0 \leq y \leq N - 1$ ) 行目には、整数  $h_{0,y}$  から整数  $h_{N-2,y}$  までの  $N - 1$  個の整数が空白区切りで与えられる。

- $2y + 2$  ( $0 \leq y \leq N - 2$ ) 行目には、整数  $v_{0,y}$  から整数  $v_{N-1,y}$  までの  $N$  個の整数が空白区切りで与えられる。
- $2N + 1$  行目には整数  $M$  が与えられる。
- 続く  $Q$  行のうち  $i$  行目 ( $1 \leq i \leq Q$ ) には、整数  $a_i$ 、整数  $b_i$ 、整数  $c_i$ 、整数  $d_i$  が空白区切りで与えられる。

## 制約

入力データは以下の制約を満たすと仮定してよい：

- $N = 200$
- $1 \leq h_{x,y} \leq 100$
- $1 \leq v_{x,y} \leq 100$
- $Q = 3000$
- $0 \leq a_i \leq N - 1$  かつ  $0 \leq b_i \leq N - 1$  かつ  $0 \leq c_i \leq N - 1$  かつ  $0 \leq d_i \leq N - 1$

## 出力

出力は、標準出力におこない、各行の末尾には改行を入れること。以下の出力形式で出力すること。

```
r1
⋮
rQ
```

各  $i$  ( $1 \leq i \leq Q$ ) 行目に頂点  $(a_i, b_i)$  から頂点  $(c_i, d_i)$  への最短の移動時間  $r_i$  を出力する。

## 入出力の例

説明の都合により、 $N$  や  $Q$  に関して前述した制約を満たさない入力例を用いる。

**例1** 以下の入力を考える。

```
3
1 9
4 4 2
3 9
9 3 1
2 3
2
0 0 2 2
0 1 0 2
```

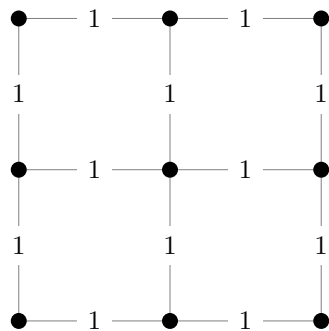
この入力の問題文中の問題例（図1）に対応する。この時、以下を出力する。

```
11
8
```

**例2** 以下の入力を考える。

```
3
1 1
1 1 1
1 1
1 1 1
1 1
4
0 0 2 0
0 1 1 0
1 0 1 0
2 0 0 2
```

この入力は以下の図に対応する。



この時、以下を出力する。

```
2
2
0
4
```

**例3** 以下の入力を考える。

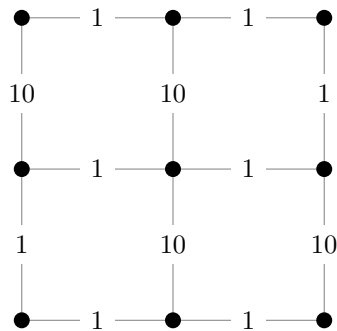
```
3
1 1
10 10 1
1 1
1 10 10
```

```

1 1
4
0 0 2 2
1 1 2 2
0 1 2 1
1 0 1 2

```

この入力は以下の図に対応する。



この時、以下を出力する。

```

8
4
2
6

```

## 2 予選問題

前節で述べた問題を解くプログラムを作成する。

### 審査方法

- 審査は、各チームの応募プログラムを、10 個の問題例に対し、各 30 秒間実行し、(i) 正解数（各問題例毎の正答したクエリ数の合計）が多い順、もし同数の場合には、(ii) 総実行時間が短い順で順位を決める。  
ただし、上記で予選通過者の選定が困難だった場合には、以下をおこなって比較する可能性がある。(a) 実行時間が 30 秒を超える問題例があった場合、実行時間の上限を変更して再実行する。(b) 総実行時間の差がつかなかった場合、入力生成器プログラム `generator.cpp` のシードを変更し、問題例の個数を増やして再実行する。
- 10 個の問題例は、入力生成器プログラム `generator.cpp` に従って生成されたものを使用する（参照：付録 A.）。
- 最終的な予選通過者の選抜には、上記の順位の外、応募時に提出するプログラムの解説 (\*1) も参考にする。(\*1) 解答プログラムの基本方針やアルゴリズムを説明した文書。

## 実行環境

- OS: macOS Ventura、CPU: プロセッサ 3.6GHz 第8世代 Intel Core i3、メモリ: 4ギガバイト以上を搭載したコンピュータ上でおこなう。
- 言語: C++ を使用する。(言語 C による提出は、C++ と同じ方法でコンパイルし、実行ファイルが生成されたプログラムのみ有効とする。)
- コンパイラは Homebrew GCC 13.2.0 を使用し、次のようにコンパイルする (ファイル名が `sc.cpp` の場合)。

```
g++-13 -O2 -std=gnu++23 sc.cpp -Wl,-ld_classic
```

実行ファイルが生成されれば、コンパイル時に warning などが出ても構わない。

- スタックサイズの上限を **65532 キロバイト** とする。具体的には、次のコマンドを実行した上で審査をおこなう：

```
ulimit -s 65532
```

## 3 級認定問題

下記の各問題を解くプログラムを作成する。

**スーパーコン3級問題** 第1節「問題説明」の問題を解くプログラムを作成する。ただし、入力は先述の制約と異なり以下の制約を満たす：

- $N = 4$
- $Q = 1$

( $N$  と  $Q$  以外に関する制約の変更はない)

**スーパーコン2級問題** 第1節「問題説明」の問題を解くプログラムを作成する。ただし、入力は先述の制約と異なり以下の制約を満たす：

- $Q = 1$

( $Q$  以外に関する制約の変更はない)

**スーパーコン1級問題** 第1節「問題説明」の問題を解くプログラムを作成する。入力に関する追加の制約はない。

## 審査方法

各級毎に用意されたいくつかの問題例に対して各 30 秒間実行し、すべてで正解を出せば合格とする。予選問題と同じ形式で出力すること。

## 実行環境

予選問題と同じ実行環境で審査する。

## 4 応募方法

SuperCon 予選の応募にあたっては、2 名ないし 3 名のチームを作り、チーム名（1 文字目が半角英字の 8 文字以内の半角英数字列）を決めること（予選通過後のチーム構成変更は認められない）。加えて、顧問もしくは指導教諭の推薦者が必要です。

級認定にのみ応募する場合、1 名で応募しても構わない（また、高校生・高専生でなくても構わない）がその場合にもチーム名はつけること。フォームに何も書かれていない場合には予選参加とみなされるので、級認定のみを希望の場合にはその旨をフォームに明記して提出すること。

以下のファイル群を **チーム名.zip** という名前で zip ファイルにして提出する。ファイルの提出先は SuperCon ウェブサイトを参照すること。

1. SuperCon 予選応募用プログラム：**チーム名.cpp**
2. プログラム実行環境の説明（実行結果、時間、使用コンパイラなど）：**チーム名-env.txt**
3. 解答プログラムの基本方針やアルゴリズムを説明した文書：**チーム名-algo.txt**、**チーム名-algo.pdf**、もしくは **チーム名-algo.docx**
4. 級認定用プログラム（予選に応募しない場合、級認定で異なるプログラムで応募する場合のみ）：
  - 1 級応募用プログラム：**チーム名-sc1.cpp**
  - 2 級応募用プログラム：**チーム名-sc2.cpp**
  - 3 級応募用プログラム：**チーム名-sc3.cpp**

SuperCon 予選に応募した場合、予選応募用プログラムを用いて 1 級、2 級、3 級の審査も同時に行われる。以下同様に、1 級に応募した場合、2 級、3 級の審査も同時に行われる。2 級に応募した場合、3 級の審査も同時に行われる。したがってそれぞれの級に対して、プログラムを提出する必要はない。

ファイルの提出先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

- チーム以外の人と解法等について相談することを禁じる。相談したことが判明した場合、本選出場を取り消すなどの措置をとることがある。
- チーム名は必ず 1 文字目が半角英字の 8 文字以内の半角英数字列であること。チーム名が不適切な場合には修正を依頼することがある。
- 例題に対して正しく動くことを確認してから応募すること。
- 実行環境やアルゴリズムの説明ファイルに利用する拡張子は使用したドキュメント作成ツールに合わせたものを利用すること。
- 本選参加者は原則全日程全員出席すること。
- 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。

- 問題解法に関してインターネットで検索することや ChatGPT 等の AI サービスを利用することは構わない。

## 付録 A. 予選問題の入力生成器

予選では、以下の入力生成器プログラム (`generator.cpp`) から 2 行目の疑似乱数のシード値 (`SEED`) を「ある値」に変更したプログラムによって生成された 10 個の問題例 (`qual_random_01.in`, ..., `qual_random_10.in`) を用いて審査する。

```
// 予選審査では SEED の値を別の「ある値」に変更する。
const unsigned long long SEED = 0x0000000000000001ull;
#include <stdio.h>
#include <stdlib.h>
#include <vector>
using namespace std;

const int N_MIN = 200;
const int N_MAX = 200;
const int C_MIN = 1;
const int C_MAX = 100;
const int Q_MIN = 3000;
const int Q_MAX = 3000;

unsigned long long xor_shift()
{
    static unsigned long long x = SEED;
    x = x ^ (x << 13);
    x = x ^ (x >> 7);
    x = x ^ (x << 17);
    return x;
}
// [0, x)
unsigned long long rnd(unsigned long long x) { return xor_shift() % x; }
// [l, r]
unsigned long long range_rnd(unsigned long long l, unsigned long long r) {
    return l + rnd(r - l + 1);
}

int main()
{
```



```

for (int t = 0; t < 10; t++)
{
    char file_name[64];
    sprintf(file_name, "qual_random_%02d.in", t + 1);

    int N = range_rnd(N_MIN, N_MAX);
    vector<vector<int>> h(N - 1, vector<int>(N));
    for (int x = 0; x < N - 1; x++)
        for (int y = 0; y < N; y++)
        {
            h[x][y] = range_rnd(C_MIN, C_MAX);
        }
    vector<vector<int>> v(N, vector<int>(N - 1));
    for (int x = 0; x < N; x++)
        for (int y = 0; y < N - 1; y++)
        {
            v[x][y] = range_rnd(C_MIN, C_MAX);
        }
    int Q = range_rnd(Q_MIN, Q_MAX);
    vector<int> a = vector<int>(Q);
    vector<int> b = vector<int>(Q);
    vector<int> c = vector<int>(Q);
    vector<int> d = vector<int>(Q);
    for (int q = 0; q < Q; q++)
    {
        a[q] = range_rnd(0, N - 1);
        b[q] = range_rnd(0, N - 1);
        c[q] = range_rnd(0, N - 1);
        d[q] = range_rnd(0, N - 1);
    }

    {
        FILE *fp = fopen(file_name, "w");
        if (fp == NULL)
            return 1;
        fprintf(fp, "%d\n", N);
        for (int i = 1; i <= 2 * N - 1; i++)
        {
            if (i % 2 == 1)

```

```

    {
        int y = i / 2;
        fprintf(fp, "%d", h[0][y]);
        for (int x = 1; x <= N - 2; x++)
        {
            fprintf(fp, " %d", h[x][y]);
        }
        fprintf(fp, "\n");
    }
    else
    {
        int y = i / 2 - 1;
        fprintf(fp, "%d", v[0][y]);
        for (int x = 1; x <= N - 1; x++)
        {
            fprintf(fp, " %d", v[x][y]);
        }
        fprintf(fp, "\n");
    }
}
fprintf(fp, "%d\n", Q);
for (int i = 1; i <= Q; i++)
    fprintf(fp, "%d %d %d %d\n", a[i-1], b[i-1], c[i-1], d[i-1]);
fclose(fp);
}
}
return 0;
}

```