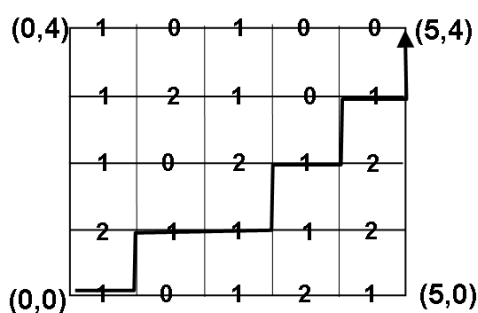


スーパーコン 2011 予選課題：級認定問題 & 予選問題

経路総数の計算：経路計算は、カーナビやウェブの経路探索サービスなどで、最近では一般にも身近な計算問題でしょう。単に一つの最適経路を求めるだけでなく、同様に良い経路をすべて求めておくことも避難計画などに重要です。

今回の予選課題は、最短経路長（もしくは準最短経路長）となる経路の総数を求める問題です。具体的には、図 1 のような $(n+1) \times (m+1)$ の格子状に表わされた道路網において、座標 $(0,0)$ のスタート格子点 S から、座標 (m,n) のゴール格子点 T までの最短経路長（もしくは準最短経路長）を達成する経路の個数を求める問題です（注：便宜上、各格子点を $(0,0)$ から (m,n) までの座標で表わしますが、この数字は辺の長さとは無関係です。また、どの問題例でもスタート頂点 S は座標 $(0,0)$ の点で、ゴール頂点は座標 (m,n) の点とします。）



左の格子図が問題例の一つ。各格子点が交差点を表わし、辺が交差点間の道路を表わしている。各辺が表わす道路の長さは一定ではない。この例では、各辺の上に書かれている整数が道路長である（縦の辺はすべて長さ 1 とする）。道路長 0 は通行できないことを意味している。

経路を考える際、交差点ではどちらに進んでもよい。この問題例では、スタート格子点 $(0,0)$ からゴール格子点 $(5,4)$ までの最短経路長は 9。その長さとなる経路の取り方は太線の矢印を含め 2 通り。

図 1．整数格子の上を表わされた道路網の例 ($m = 5, n = 4$)

以下のすべての問いにおいて、 $m, n \leq 200$ 、各辺の長さは $0 \sim 20$ の整数とします（問 A のみ、辺の長さは 0 または 1 に限る）。問題例の与え方は次のページ以降の詳細説明で述べます。また、すべての問いにおいて、答えるべき経路の総数は int に収まると仮定して構いません。

問 A（スーパーコン 3 級認定問題）

与えられる問題例の各辺の長さは 0（通行不可）もしくは 1 だけとします。この場合に、長さ $m+n$ の経路長となる経路の総数を求めるプログラムを作成してください（ヒント：長さ $m+n$ の経路は（もしあれば）最短経路です。無い場合は答えは 0 です。スタート格子点に近い各格子点 (i, j) から順に、そこまでに至る長さ $i+j$ の経路の総数を求める戦略がよいでしょう。）

問 B（スーパーコン 2 級認定問題）

与えられた問題例に対して、最短経路長とその最短経路を達成する経路の総数を求めるプログラムを作成してください。

問 C（予選問題兼スーパーコン 1 級認定問題）

問題例として、格子状道路図の他に整数 k ($1 \leq k \leq 200$) が与えられます。この問題例に対して、最短経路長、次に長い経路長、と順に考えていったときの k 番目の経路長を求め、その長さを達成する経路の総数を求めるプログラムを作成して下さい。たとえば、図 1 の例では、最短経路長の次の長さの経路長は 10 で、その経路長を達成する経路の個数は 4 なので、この格子状道路図と $k = 2$ に対する答えは 10 と 4 です。

注意

1. 作成するプログラム

- (a) プログラムは入出力の部分を規定した雛形プログラムをもとに、その一部を修正する形で作って下さい。作成の際には「変更可能」とコメントされている範囲以外に変更・削除しないようにして下さい。したがって、ここで指定された入力、出力以外のものを期待したプログラムは審査対象から除外されます。その他は適宜変更や追加して構いません。また、標準ライブラリ関数等を宣言して使用しても構いません。
- (b) 提出するプログラムは指定したファイル名の単一ファイルとして下さい。
- (c) プログラムは C 言語で記述して下さい。詳細は以下の通りです。
 - ・プログラムは C 言語規格 (ANSI C や C99 など) に準拠する C 言語で記述して下さい。(インラインアセンブラの使用は禁止します。)
 - ・int は 32 ビット, long long は 64 ビットを仮定します。long のビット幅は環境により異なるので、64 ビットデータを扱いたい場合は long ではなく long long を使って下さい。(<stdint.h> をインクルードして使える int64_t などを使うとさらに確実です。)
 - ・Linux 上の gcc ver 3.3.3 を使用してコンパイルし、Linux 上で実行します。この Linux は 64 ビット x86_64 版です。
 - ・移植性の問題 (例えばエンディアンの違い) によるトラブルには対処しません。

2. 審査方法 (スーパーコン認定に関して)¹

- (a) 応募プログラムをコンパイルし、5 題程度の問題例に対して各々 1 分間の制限時間で実行し、すべての問題例で制限時間内に正確な答えを出している場合に合格とします。
- (b) スーパーコン認定に使用する問題例では $m, n \leq 50$ (問 C の場合には $k \leq 50$) とします。
- (c) 審査環境におけるメモリはおよそ 4 ギガバイトです。

3. 審査方法 (予選選抜に関して)

- (a) 応募プログラムをコンパイルし、10 題程度の問題例に対して各々 2 分間の制限時間で実行します。
- (b) 予選選抜で使用する問題例では $m, n, k \leq 200$ とします。
- (c) 審査環境におけるメモリはおよそ 4 ギガバイトです。
- (d) 制限時間内に正確な答えを出している問題例の個数順をもとに順位を決めます。
- (e) 同順位のチームに対しては、正解を出した計算の計算時間の合計を用いて順位を決めます。(万が一、合計計算時間に優位差の無い場合には、合わせて提出されるプログラムの説明をもとに工夫度を評価して順位を決めます。)
- (f) 以上の順位付けのもとで、上位 10 チームを本選出場候補チームとして選びます。ただし、1 校からは最大 1 チームしか本選出場できません。本選出場候補チームには、同時にスーパーコン 1 級も授与します。

¹スーパーコン認定、予選選抜のどちらも審査は東工大のスパコン Tsubame2.0 の上で行う予定です。ただし (Tsubame 全体では 1 万コア以上ありますが)、審査では 1 コアのみ、かつメモリの制限を付けて実行するので、通常のパソコンと大差はありません。Tsubame2.0 のスペックについての詳細に興味のある人は <http://tsubame.gsic.titech.ac.jp/hardware-architecture> を参照してください。

問題例の与え方についての詳細説明

問題例は以下のように与えられます（ただし， k は問Cのみ）。

$m, n, k \leftarrow m, n$ は格子状道路網の横と縦の最大座標． k は何番目の長さを出力するかの指定．

$a_{0,1}, a_{0,2}, \dots, a_{0,m} \leftarrow$ 下から 1 番目の横の辺（ $(0,0), \dots, (m,0)$ 間の辺）の長さを左から

$a_{1,1}, a_{1,2}, \dots, a_{1,m} \leftarrow$ 下から 2 番目の横の辺（ $(0,1), \dots, (m,1)$ 間の辺）の長さを左から

⋮

$a_{n,1}, a_{n,2}, \dots, a_{n,m} \leftarrow$ 下から $n+1$ 番目の横の辺（ $(0,n), \dots, (m,n)$ 間の辺）の長さを左から

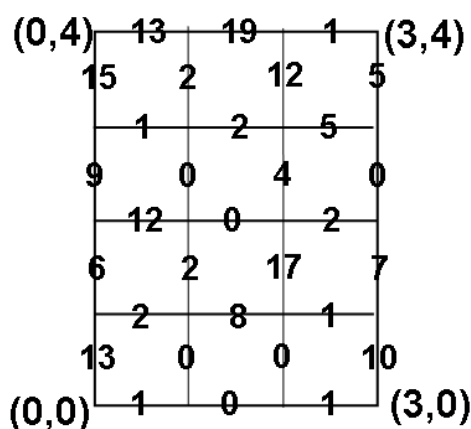
$b_{1,0}, b_{2,0}, \dots, b_{n,0} \leftarrow$ 左から 1 列目の縦の辺（ $(0,0), \dots, (0,n)$ 間の辺）の長さを下から

$b_{1,1}, b_{2,1}, \dots, b_{n,1} \leftarrow$ 左から 2 列目の縦の辺（ $(1,0), \dots, (1,n)$ 間の辺）の長さを下から

$b_{1,m}, b_{2,m}, \dots, b_{n,m} \leftarrow$ 左から $m+1$ 列目の縦の辺（ $(m,0), \dots, (m,n)$ 間の辺）の長さを下から

たとえば右図の格子状道路網に対する問Aのプログラムに対する入力は次のようになります。

3, 4
 1, 0, 1
 2, 8, 1
 12, 0, 2
 1, 2, 5
 13, 19, 1
 13, 6, 9, 15
 0, 2, 0, 2
 0, 17, 4, 12
 10, 7, 0, 5



雛形プログラムについて：

雛形プログラムでは，上記の入力問題例の辺長のデータを，次のような配列 $D[m+1][n+1][4]$ に格納するまで行います．それを用いて計算してください．

$D[0][0][0] = 13 \leftarrow$ 北方向
 $D[0][0][1] = 1 \leftarrow$ 東方向
 $D[0][0][2] = 0 \leftarrow$ 南方向（辺がないので 0）
 $D[0][0][3] = 0 \leftarrow$ 西方向（辺がないので 0）
 $D[1][0][0] = 0$
 $D[1][0][1] = 0$
 $D[1][0][2] = 0$
 $D[1][0][3] = 1$
 ⋮
 $D[2][3][0] = 12$
 $D[2][3][1] = 5$
 $D[2][3][2] = 4$
 $D[2][3][3] = 2$
 ⋮