

## ACM ICPC2013 国内予選 [問題 A 整長方形](#)

### 【問題と考察】

- 与えられた横長整長方形に対して、それより大きく最小の横長整長方形を求める
- 横長整長方形の大小関係
  1. 対角線の長さが短い方が小さい
  2. 対角線の長さが同じなら、高さの低い方が小さい
- 対角線の長さの比較は、誤差の影響を避けるため、対角線の長さの二乗で行えば良い
- 求める横長整長方形は、高さ幅共に 150 以下であることが保証されている
- 高さと幅が 150 以下の横長整長方形の個数は高々  $150 \times 149 / 2 = 11,175$  個なので、高さと幅が 150 以下の全ての横長整長方形を与えられた横長整長方形と比較する方法が良い

### 【プログラム例】

```
// ACM ICPC2013 Domestic Problem A
// http://sparth.u-aizu.ac.jp/icpc2013/d_problems.php?lang=jp#section_A
// Filename:      pa.c
// Compile:       cc pa.c
// Execution:     ./a.out < A0 > A0.result
// Check:        diff A0.result A0.ans
// Algorithm:
//      高さ と 幅 が 1 5 0 以下 の 全 て の 横 長 整 長 方 形 を 順 に チェック し
//      与 え ら れ た 横 長 整 長 方 形 より 大 き い 最 小 の 横 長 整 長 方 形 を 求 め る

#include <stdio.h>
#define MAX_HW 150           // 高さ と 幅 の 最 大 値

// 長 方 形 1(h1,w1)の ほう が 長 方 形 2(h2,w2)より 小 さ け れ ば 1, さ も な く ば 0 を 返 す 関 数
int is_small(h1, w1, h2,w2)
{
    int d1, d2;
    d1 = h1*h1+w1*w1;         // 長 方 形 1 の 対 角 長 の 二 乗
    d2 = h2*h2+w2*w2;         // 長 方 形 2 の 対 角 長 の 二 乗
    if(d1 < d2) return 1;     // 対 角 長 が 短 け れ ば 小 さ い
    else if(d1 > d2) return 0; // 対 角 長 が 長 け れ ば 大 き い
    else if(h1<h2) return 1;  // 対 角 長 が 等 し い と き は 高 さ の 低 い 方 が 小 さ い
    else return 0;
}

int main()
{
```

```

int h,w;          // 与えられた長方形の高さ、幅、対角長の二乗
int mh, mw, md;  // 求める長方形の高さ、幅、対角長の二乗
int ch, cw, cd;  // 現在チェック中の長方形の高さ、幅、対角長の二乗

while(1){
    scanf("%d %d", &h, &w);          // 高さとの幅の入力
    if(h==0 && w==0) break;          // 共に 0 なら終了
    mh = mw = MAX_HW+1; // とりあえず求める長方形の高さと幅は最大値+1 に初期化
    for(ch = 1; ch <= MAX_HW; ch++){
        for(cw = ch+1; cw <= MAX_HW; cw++){ // 横長長方形の生成
            if(ch == h && cw == w) continue; // 同じ長方形はチェックしない
            if(is_small(ch, cw, h, w)) continue; // 小さい長方形は無視
            // この長方形は与えられた長方形より大きい
            if(is_small(ch, cw, mh, mw)){ // 今まで求まっているものより小さいので
                mh = ch; mw = cw; // 答えを更新
            }
        }
    }
    printf("%d %d¥n", mh, mw);
}
}

```