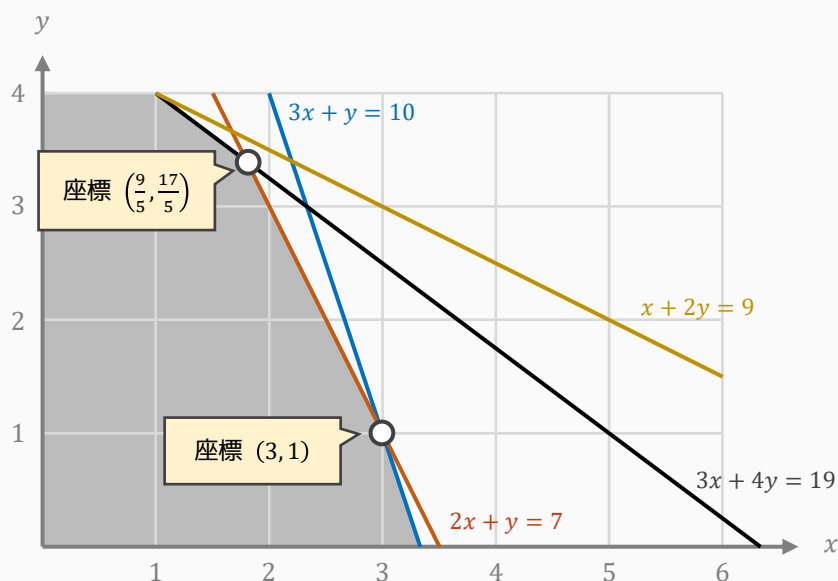
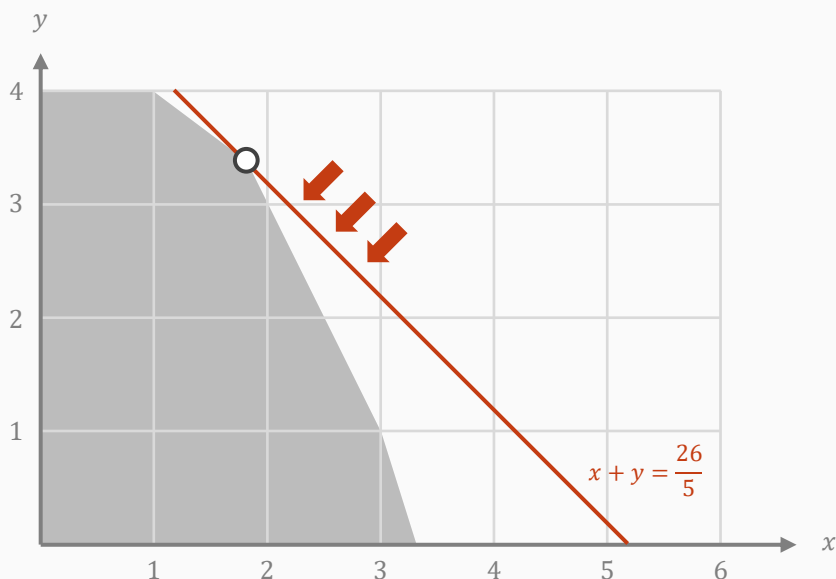


問題 5.5.1

まず、 $3x + y = 10$, $2x + y = 7$, $3x + 4y = 19$, $x + 2y = 9$ のグラフを同じ座標平面上に描くと以下ようになります。灰色は問題文の 4 つの条件をすべて満たす領域を示しています。

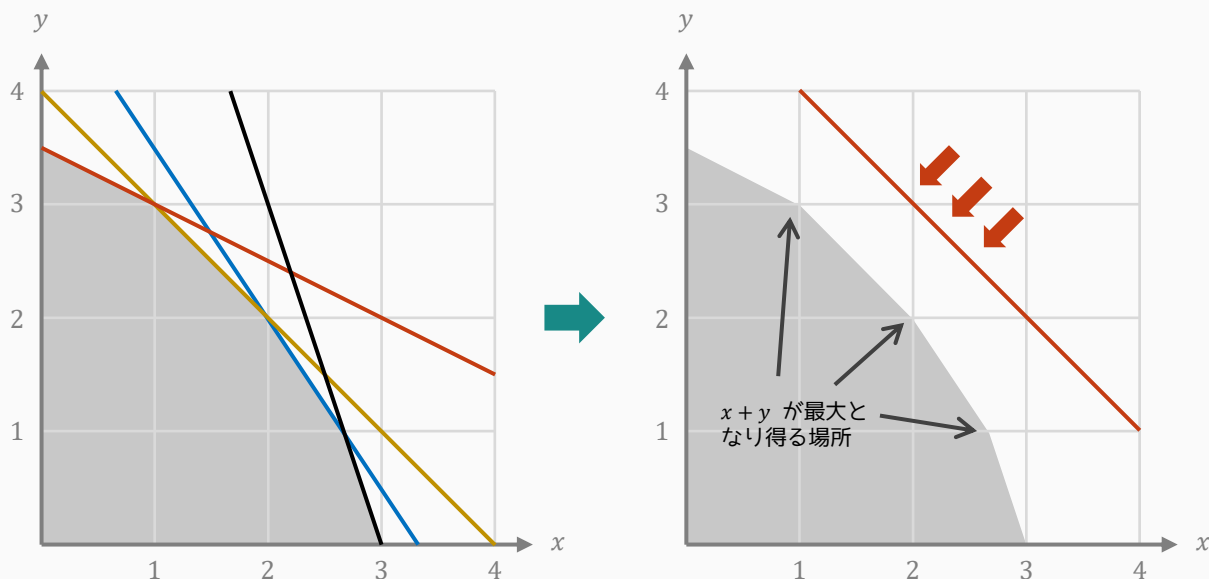


そこで、直線 $x + y = a$ を少しずつ下ろしてみましょう (a を小さくしてみましょう)。そうすると、 $a = 26/5$ のときに初めて灰色部分とぶつかるので、 $x + y$ の最大値は $26/5$ です。



問題 5.5.2

線形計画問題では、2つの直線の交点で $x + y$ が最大となります。なぜなら、条件を満たす部分（下図の灰色に相当）は、2直線の交点でしか曲がらないからです。



したがって、以下のようなアルゴリズムによって答えを求めることができます。

なお、 x, y が正の実数のときに最大値をとることが制約で保証されていることに注意してください。（そうでないケースでは最大値が無限大になる可能性があり、場合分けが必要です）

すべての整数の組 (i, j) [$1 \leq i < j \leq N$] について、以下のことを行う。

- 直線 $a_i x + b_i y + c_i$ と直線 $a_j x + b_j y = c_j$ の交点を求める
- N 個の条件式すべてを満たすかどうかを判定する

条件を満たす交点の中で $(x \text{ 座標}) + (y \text{ 座標})$ の値が最大となるものが答え。

なお、直線 $a_i x + b_i y + c_i$ と直線 $a_j x + b_j y = c_j$ の交点は以下の方法で求められます。

- $a_i b_j = a_j b_i$ のとき：2直線は平行（交差しない）
- $a_i b_j \neq a_j b_i$ のとき：交点は以下の座標となる（本書では扱っていない連立方程式を使うと使うと導出できます。興味のある人は調べましょう）

$$\left(\frac{c_i b_j - c_j b_i}{a_i b_j - a_j b_i}, \frac{c_i a_j - c_j a_i}{b_i a_j - b_j a_i} \right)$$

したがって、以下のようなプログラムを書くと正解が得られます。関数 `check(x, y)` は実数の組 (x, y) が N 個の条件式をすべて満たす場合 `true`、そうでない場合 `false` を返します。

なお、このプログラムは `check` 関数を $O(N^2)$ 回呼び出しており、`check` 関数の計算量は $O(N)$ であるため、プログラム全体の計算量は $O(N^3)$ となります。

```
#include <iostream>
using namespace std;

int N;
double A[509], B[509], C[509];

bool check(double x, double y) {
    for (int i = 1; i <= N; i++) {
        if (A[i] * x + B[i] * y > C[i]) return false;
    }
    return true;
}

int main() {
    // 入力
    cin >> N;
    for (int i = 1; i <= N; i++) cin >> A[i] >> B[i] >> C[i];

    // 交点を全探索
    double Answer = 0.0;
    for (int i = 1; i <= N; i++) {
        for (int j = i + 1; j <= N; j++) {
            // 交点をもたない場合
            if (A[i] * B[j] == A[j] * B[i]) continue;

            // i 番目の直線（条件式の境界）と j 番目の直線（条件式の境界）の交点を求める
            double px = (C[i] * B[j] - C[j] * B[i]) / (A[i] * B[j] - A[j] * B[i]);
            double py = (C[i] * A[j] - C[j] * A[i]) / (B[i] * A[j] - B[j] * A[i]);
            bool ret = check(px - 0.00000001, py - 0.00000001);
            if (ret == true) {
                Answer = max(Answer, px + py);
            }
        }
    }

    // 答えを出力
    printf("%.12lf\n", Answer);
    return 0;
}
```

※ Python などのソースコードは chap5-5.md をご覧ください。