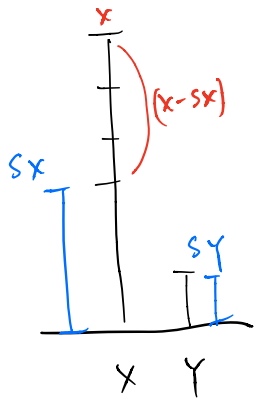
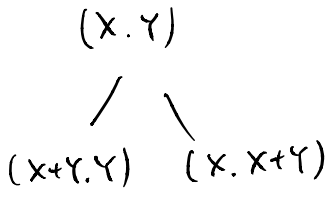



```
function reachingPoint(sx, sy, tx, ty) {
    let x = tx, y = ty;
    while(x >= sx && y >= sy) {
        if(x > y) {
            if(y === sy) {
                return (x-sx) % y === 0;
            }
            x = x % y;
        } else {
            if(x === sx) {
                return (y-sy) % x === 0;
            }
            y = y % x;
        }
    }
    return false;
}
```



for a given number, it comes from two cases.

$$\begin{cases} (x.y) \swarrow (x-y, y) \\ \qquad\quad \searrow (x, y-x) \end{cases}$$

also $x.y \geq 0$, it must come from a positive number.

So. if $x > y$, it comes from $(x-y, y)$

else $(x. y-x)$.

So whenever $x >= y$, we keep substracting.

which is equivalent to $x \% y$.