

D 玩具制造商

时间限制：2000ms 内存限制：65536kb

通过率：2/53 (3.77%) 正确率：2/189 (1.06%)

题目描述

妮妮是一个玩具制造商，她手下有一个玩具加工厂。加工厂每次可以从以下的两种方法选择一种进行玩具加工。

- x 个零件 A 与 y 个零件 B 加工成一个玩具。
- y 个零件 A 与 x 个零件 B 加工成一个玩具。

妮妮的加工厂目前共有 a 个零件 A 与 b 个零件 B，妮妮想知道加工厂最多可以加工出多少个玩具。

输入

第一行一个正整数 T ($1 \leq T \leq 5 \times 10^5$)，表示数据组数。

接下来 T 行，每行四个正整数 x, y, a, b ($0 \leq x, y, a, b \leq 10^9$)，含义如题，保证 x, y 不同时为 0。

输出

输出 T 行，第 i 行一个整数，表示第 i 组数据最多可以加工的玩具个数。

输入样例

```
1
1 2 3 3
```

输出样例

```
2
```

样例解释

第一个样例使用 1 个零件 A 和 2 个零件 B 加工成一个玩具，再使用 2 个零件 A 和 1 个零件 B 加工成一个玩具，共获得 2 个玩具。

这题首先需要满足两个不等式，假设第一种方法制造 k_1 个玩具，第二种方法制造 k_2 个玩具

| 数量 | a | b |
|-------|-----|-----|
| k_1 | x | y |
| k_2 | y | x |

$$\begin{aligned}k_1 * x + k_2 * y &\leq a \\k_1 * y + k_2 * x &\leq b \\(k_1 + k_2) * (x + y) &\leq a + b \\num = (k_1 + k_2) &\leq \frac{a + b}{x + y}\end{aligned}$$

根据题目数据范围 $0 \leq x, y, a, b \leq 10^9$,num数值可能达到 $2 * 10^9$

因此我们使用二分搜索答案

每个二分的值表示当前check能否制造这么多数量的玩具

如果能制造，我们就搜索更大数值，不能制造，就搜索更小数值

```
1 long long max = (long long) ((a + b) / (x + y));
2 long long l = 0, r = max;
3 while (l <= r) {
4     long long mid = l + (r - l) / 2;
5     int k = checkTotal(mid);
6     if (k == 1) l = mid + 1;
7     else r = mid - 1;
8 }
9 printf("%d\n", l - 1);
```

由于这里的max最大可能为 $2 * 10^9$,那么计算 $(l+r)/2$ 的过程可能会超出int 范围，这里可以采用long long，或者用 $l+(r-l)/2$ 来计算mid

接下来是核心部分，检查某个数值的玩具能否被制造

如果 $x=y$ ，那么我们直接计算即可

如果 $x \neq y$ ，我们考虑如下图表，这里假设 $x > y$

| 总数量num | a | b |
|--------|---------------|---------------|
| k1 | $(x - y) + y$ | y |
| k2 | y | $(x - y) + y$ |

不论k1和k2如何分配num

制造num个玩具一定消耗 $\text{num} * y$ 个A物品，消耗 $\text{num} * y$ 个B物品

此时剩余的A物品全部补齐用于制造第一类的数量，也就是 $\frac{a}{x-y}$

剩余的B物品全部补齐用于制造第二类的数量，也就是 $\frac{b}{x-y}$

这样我们得到的 $\frac{a+b}{x-y}$ 即为当前情况下能制造的最大数量，如果它大于等于num，那就说明当前这个数量是可以制造的

由于最开始规定了 $x > y$ ，因此我们计算时替换成 $|x-y|$ 即可，最开始的消耗替换为 $\min(x,y) * \text{num}$

```
1  int checkTotal(long long num) {
2      long long A = a;
3      long long B = b;
4      long long p = min(x, y);
5      p *= num;
6      A -= p;
7      B -= p;
8      if (A >= 0 && B >= 0) {
9          if (x != y) {
10             long long tmp = A / 1llabs(x - y) + B / 1llabs(x - y);
11             if (tmp >= num) return 1;
12             else return 0;
13         } else return 1;
14     } else return 0;
15 }
```