

狼羊过河

时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

一农夫带着 m 只羊， n 只狼过河，农夫有一条可载 x 只狼/羊的船；农夫在时或者羊的数量大于狼时，狼不会攻击羊；农夫在不损失羊的情况下，运输几次可以完成运输？(返程不计入次数)

输入描述：

输入参数为 m, n, x ;
 m 为羊的数量、 n 为狼的数量、 x 为可载狼和羊的数量

输出描述：

返回运输次数即可

补充说明：

如果无法完成运输返回0;

示例1

输入：

5 3 3

输出：

3

详解：

第一次：2只狼

第二次：三只羊

第三次：2只羊，1只狼

解题思路：

通过递归来模拟出狼羊过河的所有情况，找出其中次数最少的。

```

public class Main{

    public static int min = Integer.MAX_VALUE;
    public static int countY;    //羊的总数
    public static int countL;    //狼的总数

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int m = sc.nextInt();
        int n = sc.nextInt();
        int x = sc.nextInt();

        countY = m;
        countL = n;

        guohe( m, n, x,0);

        if(m + n <= x){        //一趟能运完
            System.out.println(1);
        }else if(min == Integer.MAX_VALUE){
            System.out.println(0);
        }else {
            System.out.println(min);
        }

    }

    /**
     *
     * @param m    岸边的羊的个数
     * @param n    岸边的狼的个数
     * @param x    船的承重
     * @param count    过河的次数
     */
    public static void guohe(int m, int n,int x,int count){

        if(m + n <= x){        //剩下的能一次运完
            min = Math.min( min, count+1);
        }else {
            for(int i=0; i<=m; i++){        //过河的羊的个数
                for(int j=0; j<=n; j++){        //过河的狼的个数
                    if((i+j == 0) || (i+j >x)){        //在船上的狼羊总数需要大于 0 且小于等

```

于 x

```
continue;
```

```
}
```

```
if(m-i != 0 && m - i <= n - j){ //剩下的羊在不为 0 的情况下必须大
```

于狼

```
continue;
```

```
}
```

```
if(countY-(m-i) != 0 && (countY-(m-i)) <= (countL-(n-j))){ //对岸的  
羊在不为 0 的情况下必须要大于狼
```

```
continue;
```

```
}
```

```
guohe(m-i,n-j, x,count+1);
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```