

Java方向编程题答案

####

day10

[编程题]45846-Fibonacci数列

链接: <https://www.nowcoder.com/questionTerminal/18ecd0ecf5ef4fe9ba3f17f8d00d2d66>

【题目解析】：

Fibonacci数我们之前在作业中都做过了，只要大家掌握了之前Fibonacci数列的构造方法，这道题目做起来应该难度不大。

【解题思路】：

首先构建一个Fibonacci数列，然后从该数列中找到与n最接近的值 在构建数列的时候，如果使用递归，就会因重复计算产生成倍的时间代价，所以用数组的形式 即：A[n]=A[n-1]+A[n-2] 但是数组又会产生额外的空间花销，考虑到每次迭代只会用前两项的值，所以我们可以只用三个变量 a b c分别表示每次迭代中的 A[n] A[n-1] A[n-2] 计算数列的过程中，如果发现某一项大于等于n，则迭代就可以停止了。此时n的值介于最后的两项之间。这时候只要找到n和这两项距离最小者即可。

【示例代码】：

```
import java.util.*;
public class Main {
    public static void main(String[] args) throws Exception{
        Scanner scanner = new Scanner(System.in);
        int n = 0;
        int a = 0,b = 1,c = 1;
        if (scanner.hasNextInt()) {
            n = scanner.nextInt();
        }
        while (c < n) {
            a = b;
            b = c;
            c = a + b;
        }
        int dis1 = n - b;
        int dis2 = c - n;
        System.out.println(dis1<dis2?dis1:dis2);
    }
}
```

[编程题]24965-机器人走方格I

链接链接: <https://www.nowcoder.com/questionTerminal/e8bb8e68434e42acbcdff0341f2a32c5>

【题目解析】：

题目本身还是挺简单的，由于每一步只有两种选择，因此采用递归即可。而且题目已经约束了 $x + y < 12$ ，因此也不用担心递归深度的问题。

【解题思路】：

机器人在XY的矩阵中走，每一步都有两种选择：要么向下、要么向右。如果向下走，问题就变成：求(X-1)Y矩阵中机器人的走法；如果向右走，问题就变成：求X*(Y-1)矩阵中机器人的走法；显然是递归的思想！既然是递归，再考虑退出条件：当整个矩阵只有一行 或 一列的时候只有一种走法。

【示例代码】：

```
public class Robot {  
    public int countWays(int x, int y) {  
        if ( x==1 || y==1 )  
            return 1;  
        return countWays(x-1,y)+countWays(x,y-1);  
    }  
}
```

#####

比特科技制作