

例题

```
void printPairs(int[] array) {  
    for (int i = 0; i < array.length; i++) {  
        for (int j = i + 1; j < array.length; j++) {  
            System.out.println(array[i] + ", " + array[j]);  
        }  
    }  
}
```

时间复杂度计算 – 递归问题

什么是递归：在数学与计算机科学中，是指在函数的定义中使用函数自身的方法。递归一词还较常用于描述以自相似方法重复事物的过程。

```
int calculate(int n) {  
    if (n <= 0) {  
        return 1;  
    }  
    return calculate(n - 1) + calculate(n - 1);  
}
```

经验性结论：
递归问题的时间复杂度通常（并不总是）看起来形如 $O(\text{branches}^{\text{depth}})$
其中branches指递归分支的总数，
depth指递归调用深度