



# Java Lab3: 聊聊别的

- 目 题目讲解
- M Markdown
- </> 杂项



# 题目讲解

*you want it, you search it.*

讲题之前，首先介绍两个概念：

## 1. STFW, aka Search The F\*\*king Friendly Web

- 作为现代大学生的生存必备技能
- 你有很多搜索引擎，比如必应，谷歌，百度
- 你有很多论坛，比如stackoverflow, csdn, 百度知道, 百度贴吧
- 你有很多百科，比如维基百科, 百度百科

搜索的关键在于搜索到关键。

### ■ 搜索的技巧？

- Advanced Search
- Key Word
- Use English If possible

# 题目讲解

*you want it, you read it.*

## 2. RTFM, aka Read The F\*\*king Friendly Manual

- 一个经典笑话

如果你对我的代码有疑问，请去阅读文档。

如果你依然对我的代码有疑问，请仔细阅读文档。

如果你还是对我的代码有疑问，请熟读并背诵文档。

- 文档定义了的就是对；
- 文档没定义的可能是对的，但它一定是未定义的；
- 建议不要钻牛角尖。
- 怎么找文档？STFW。
- Java文档
- 一个镜像

# 题目讲解

*you want it, you try it.*

举个栗子：

7. 给定5个整数 (通过命令行参数), 找出它们的中位数 (即第3大的数).

- 什么是命令行参数？我听说有个东西叫Scanner，是命令行参数吗？

甚么是命令行？STFW。

- 排序要我手动写吗？

当然可以，不过建议试试RTFM和STFW。

当你觉得一个功能非常常用，应该被实现的时候，它大概率已经被实现了。

```
public class Main{
    public static void main(String[] args)
    {
        int a[] = new int[5];
        // get your command line arguments.
        // sort your array.
        System.out.println(a[2]);
    }
}
```

# 题目讲解

*you want it, you recurse it.*

Hadamard 矩阵 $H(N)$ 为 $2^{N-1} \times 2^{N-1}$ 的方阵. 矩阵中的元素为 0 或者 1.  $H(1) = [1]$ ,

$$H(N) = \begin{bmatrix} H(N-1) & H(N-1) \\ H(N-1) & \neg H(N-1) \end{bmatrix}$$

其中  $\neg H(N-1)$  表示将 $H(N-1)$ 中的0变成1, 1变成0. 例如:

$$H(2) = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, H(3) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

给定  $N$ , 请输出  $H(N)$ .

- 一眼定阵, 鉴定为递归
- 递归怎么写?

要理解递归, 首先要理解什么是递归。

C语言怎么写递归, Java就怎么写递归。

# 题目讲解

*you want it, you recurse it.*

递归的两个核心问题：

## 1. 怎么递归

首先，考虑如何将大问题转化为同类型的规模更小的问题；  
然后，考虑如何在小问题已经解决的假设前提下解决当前问题。

## 2. 何时结束

比如：可以在问题规模足够小的时候结束。

套用到本题：

## 1. 怎么递归

题目暗示很明显，根据 $H(N)$ 和 $H(N - 1)$ 之间的关系递归。

## 2. 何时结束

$H(1) = [1]$ ，规模足够小，足够简单。

*P.S.: 原则上讲任何递归能解决的问题，循环都能解决。但是一般这样的问题用递归会更方便（并不是更好），并且对于复杂的问题，可能需要借助一些数据结构（如栈）。*

# 题目讲解

*you want it, you implement it.*

Alice 去参加一个聚会. Bob 也在这个聚会中. 看到 Alice 之后, Bob 把一个和 Alice 有关的八卦谣言告诉了他的一个同伴. 随后谣言开始在聚会中传播. 假设人们都按照以下方式传递谣言: 如果他第一次听到, 则从其他的人中随机选择一个 (除了告诉他的那个人和Alice), 将谣言传递出去. 如果他已经知道了这个谣言, 那么他停止传播. 请通过模拟来估计在谣言传播停止前, 所有人(除了 Alice)都知道这个谣言的概率. 同时, 请估计听到谣言人数的期望值.

- 简单模拟执行。
- 数学模型？

# 题目讲解

*you want it, you arrange it.*

## ■ 排列

- 套for循环思路？选一个，再选一个，再选一个...
- 是否可行？
- 那怎么办？递归&回溯

## ■ 置换

简单索引，两层循环。

## ■ 蛇形矩阵

一圈又一圈

循环控制

或者，模拟走迷宫？



# 来学学Markdown!

- 甚么是Markdown
- 语法

# 甚么是Markdown

*you want it, you mark it.*

*Markdown is a lightweight markup language for creating formatted text using a plain-text editor.*

- 轻量级的，格式化的，好看的，简单的，推荐的。
- word不好吗？

好和不好，如何评判？

Same question: word适合用来写代码吗？

- 应用很广泛，不会写markdown等于去长沙没吃臭豆腐
- **真的很简单，真的很方便。**
- 写报告，写文档，（至少我们这门课）推荐尝试用Markdown写文档。

# 语法

*if it can be called "grammar"*

- 其实只是简单标记(mark)，和平时在课本上画的圈圈星星三角形一个道理。

- 常用的只有几个：

- 标题：`#`
- 加粗斜体：`\*\* \*\*\*\* \*\*\*\*\*`
- 列表：`- + \*`
- 代码块：``
- 分割线：`---`

- Markdown语法规则有很多，视不同的解释器而言。但基本上会遵循最基础的一套语法。

# </> 杂项

- 代码风格
- 命名规范
- 注意事项

# 杂项

## ■ 代码风格

- 缩进
- 括号
- 格式化文档

## ■ 命名规范

- 驼峰命名法
- 下划线命名法
- 随意命名法⊙

## ■ 注意事项

- 不要提交`.class`文件
- 压缩包的命名规范：实验编号\_学号\_姓名.zip (例如: Lab1\_10215102000\_陈东.zip)
- 尽量别交txt上来，如果要写README建议用Markdown.