Java04-控制流

Task1

```
//这个函数用于判断传入的年份是否为闰年
//是闰年返回1，不是闰年返回2
int isLeapYear(int year){
 if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
        return 1;
    } else {
        return 2;
    }
    }
    //感觉题目boolean类型应该返回不了整数,所以我把boolean改成了int
```

首先说明语法糖的意思:简化语法结构

我认为switch-case 不是 if-else 的简单语法糖。它有独立的语法和更高效的底层实现，尤其在分支较多时性能更优。

switch-case

适用于对某个变量的多个固定值进行分支判断（如 int、char、String 等）。 底层实现通常会编译为"查找表"或"跳转表"（tableswitch/lookupswitch 指令），效率高，分支多时性能优于 if-else。 代码可读性好，结构清晰。

if-else

适用于任意复杂条件判断（可以是区间、逻辑表达式等）。 底层实现为一系列条件判断和跳转，分支多时效率低于 switch。

Task2.for-while

代码如下:

```
void print(int n){
 int mid = n / 2;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (j == mid - Math.abs(mid - i) || j == mid + Math.abs(mid - i))
{
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
}
```

思路:首先根据对称性,我联想到了绝对值的运用; 接着由于每行代码都要打印n个字符,故使用嵌套循环且i与j都循环n次 最后选择了最笨的方法:找规律,找完在总结浓缩n

Task4

版本一---递归

```
int Fibonacci(int n){
    if (n==1) return n;
    return Fibonacci(n-1)+ return Fibonacci(n-2)
}
```

版本二---迭代

```
int Fibonacci(int n){
     if (n==1) return n;
     int a = 0, b = 1;
    for (int i = 2; i <= n; i++) {
        int temp = a + b;
        a = b;
        b = temp;
    }
    return b;
}
```

(科学询问网络)

迭代和递归的不同主要在于:

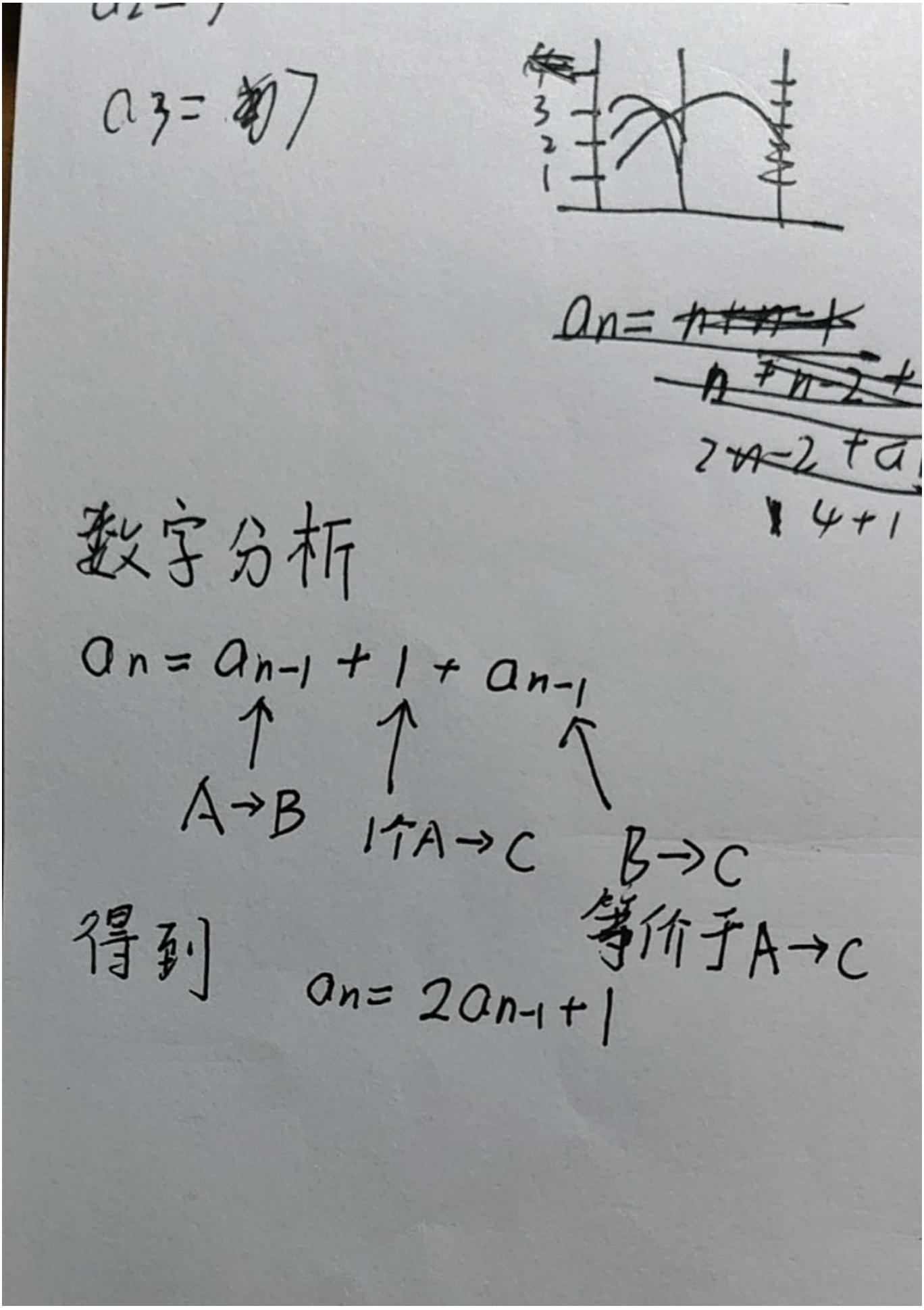
迭代是通过循环结构（如 for、while）反复执行代码块，通常用变量保存中间结果，效率高，内存消耗小。 递归是函数自己调用自己，每次调用会在栈上保存现场，代码简洁但效率低，递归层数深时容易导致栈溢出。 一般偏好迭代的原因:

迭代效率更高，不会有栈溢出风险，适合大数据量和高性能场景。 递归虽然表达简洁，但每次递归都要消耗额外的栈空间，容易导致性能问题。 循环是否能完全用递归取代?

理论上可以，任何循环都能用递归实现（因为递归和循环本质上都能实现重复计算）。 但实际开发中，递归会带来额外的函数调用开销和栈空间消耗，所以只有在问题本身具有递归结构时才推荐递归，其他情况更推荐迭代。

Task4

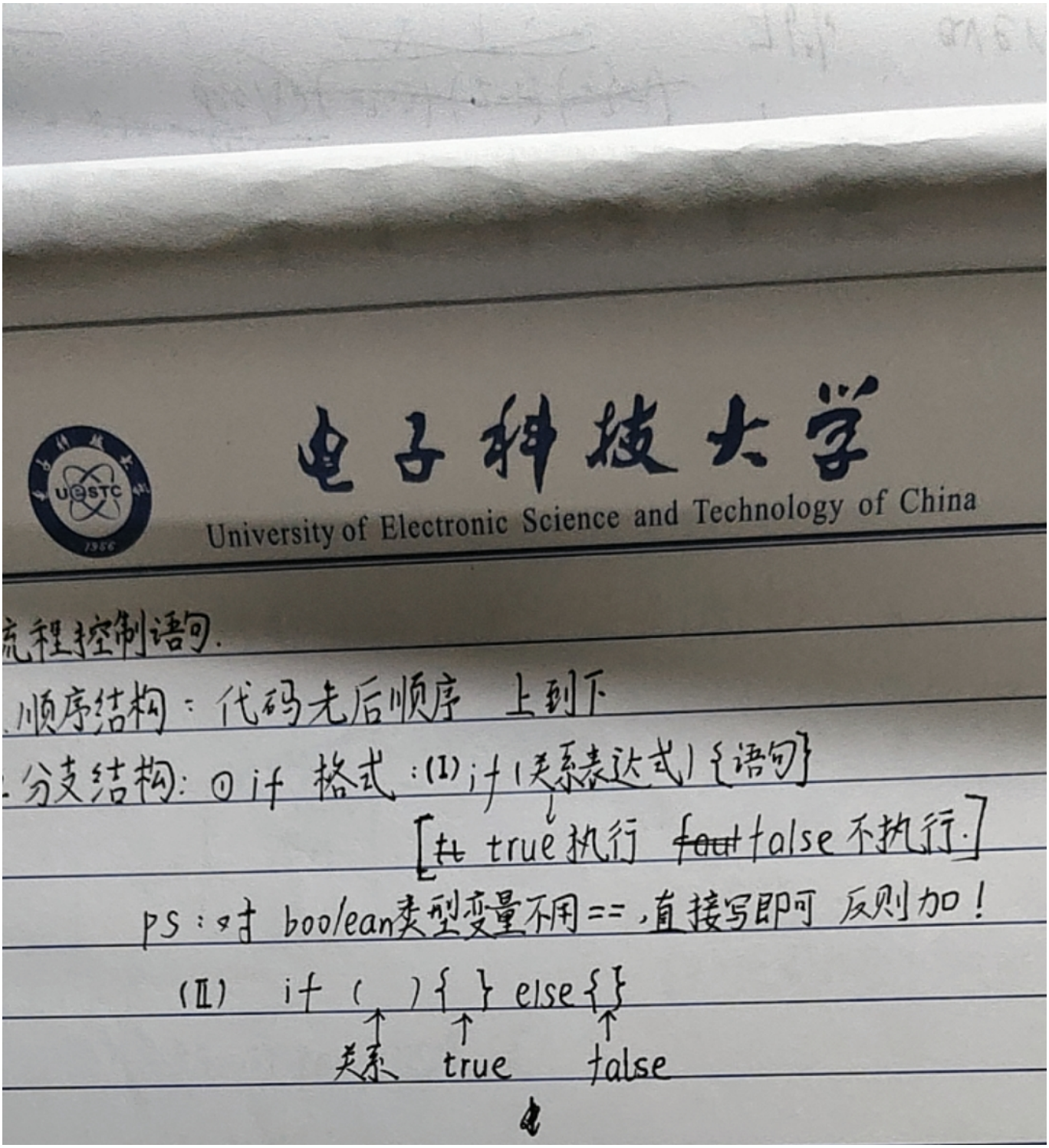
PS:这题高中做过,立马联想到了递归来解决 数学数列方法如图:



```
void hanoi(int n,char a,char b,char c){
     if (n==1) {
            System.out.println(a+"->"+c);
            return;
        }else {
                move(n-1,a,c,b);//将n-1个盘子从a挪到b
            System.out.println(a+"->"+c);//将最底层盘子从a挪到c
                move(n-1,b,a,c);//将n-1个盘子从b挪到c,其实等价于从a挪到c,将a换成b即可

        }
}
```

附加一些笔记



(三) if ( ) {} else if () {} ..... else {}

② switch ~~{ }~~

格式： switch (表达式) { case 值1： ~~case 值2~~

case 值1：

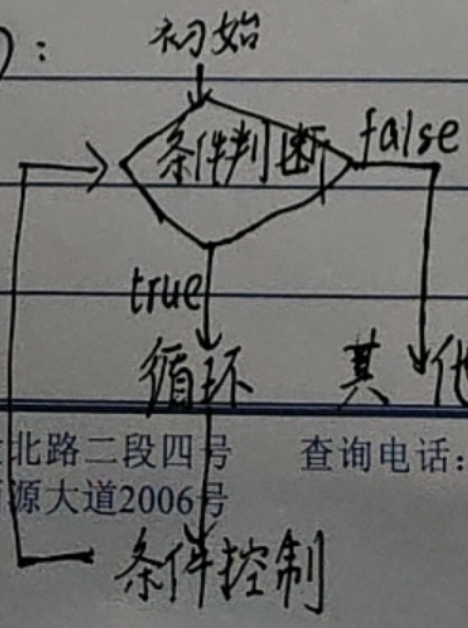
语句体1；

break； ..... default：_ ；break；}

循环结构：① for 格式 for( ) { }

初始条判条控 循环体语句

图：



eg：for(int i=1;i<=10;i++)

~~{ sout "HelloWorld" }~~

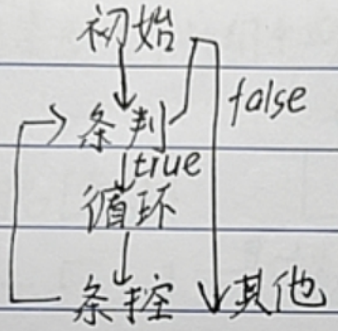
① 变量只在大括号中生

② 变量若在循环里，

：成都市建设北路二段四号 查询电话：83201114 邮政编码：610054（沙
高新西区西源大道2006号 611731（清

电子科技大学
University of Electronic Science and Technology of China

② while 格式 while(条判){循环;条控}



区别：变量是否会消失（写内

① 控制循环的

② 知范围 → for

不知范围知条件 → while

PS：用 temp 记录值！

转控制

若要~~条~~过跳：continue（一次），break结束

Random 获取随机数：

import java.util.Random;

Random r= new Random();

int number = r.nextInt(范围) bound：xx 从0开始到

(I) 无限循环

(II) 嵌套循环△ for里再来for