



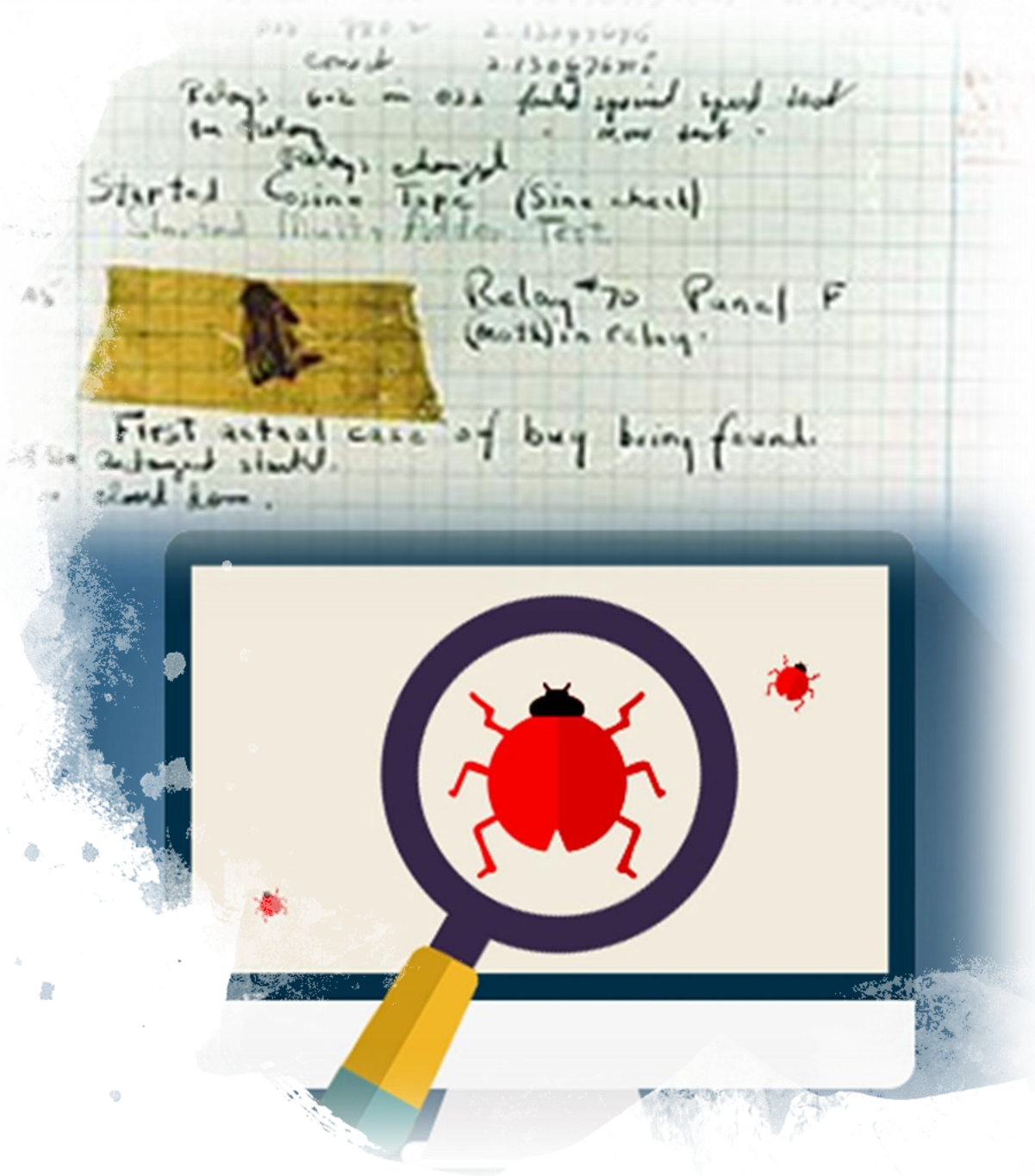
亲子算法课 (6)

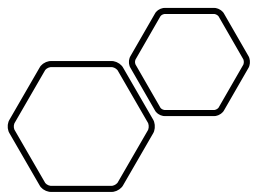
——Bug和Debug

作者：叶蒙蒙

什么是Bug

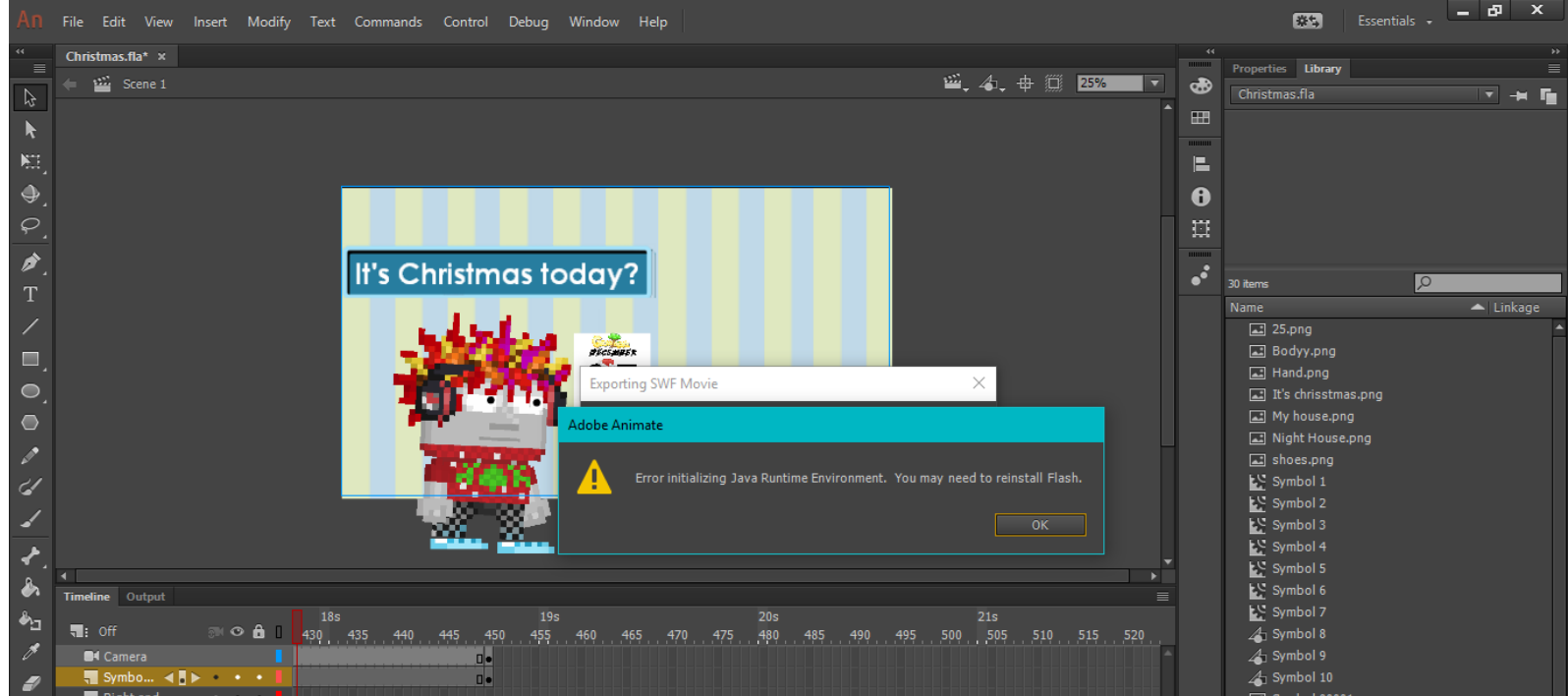
- 一只蛾子
- “么蛾子”
- 程序的错误
 - 任何情况下都不能输出正确结果
 - 大多数情况下可以输出正确结果
- 和程序形影不离的冤家





什么不是Bug

- 语法错误
- 运行环境或配置导致的程序无法运行



Bug产生的原因

- 程序的编写者逻辑不清
- 对语言掌握不够
- 对工具和运行环境了解不够



一起来看一个Bug

- 二分查找报错!

```
def binary_search(arr, target_number):  
    low = 0  
    high = len(arr) - 1  
    while (low < high):  
        middle = int((high - low) / 2 + low)  
  
        if (arr[middle] == target_number):  
            return middle  
        else:  
            if (arr[middle] < target_number):  
                low = middle + 1  
            else:  
                high = middle - 1  
  
    return -1  
  
if __name__ == "__main__":  
    arr = []  
    i = 1  
    while (i < 11):  
        arr.append(i)  
        i = i + 1  
  
    r = binary_search(arr, 4)  
    if (r == -1):  
        print ("failed")  
    else:  
        print ("succeed to find {0}".format(r))
```

- 运行结果——“failed”

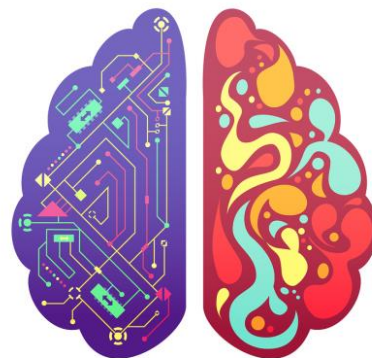


Debug & Fix bug

- 亦步亦趋找bug
- 找到它，修改它
- 修改之后不影响其他功能



如何写出没有Bug的程序



- 逻辑清晰
- 测试
 - 测试用例的选择——corner case
 - 随时debug, fix bug
 - 回归测试





如何才算学会了一个算法？

由浅入深掌握算法 (1)

听说

- 知道算法名
- 知道算法功能

了解

- 知道算法原理（自然语言描述）
- 知道算法优缺点

理解

- 知道算法的过程和细节
- 能绘制算法流程图
- 知道算法的时空复杂度

由浅入深掌握算法 (2)

实现

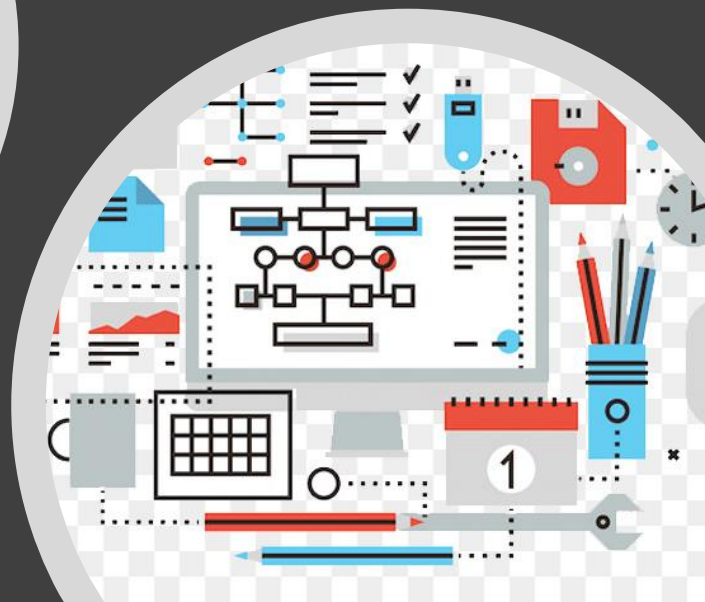
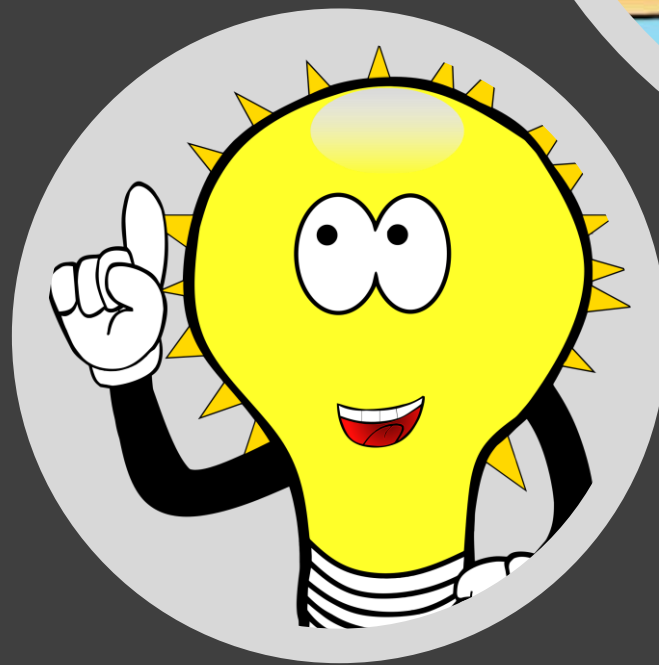
- 用编程语言编写出无bug的算法

应用

- 在解决实际问题中应用算法

以二分查找为例，看 看如何掌握算法

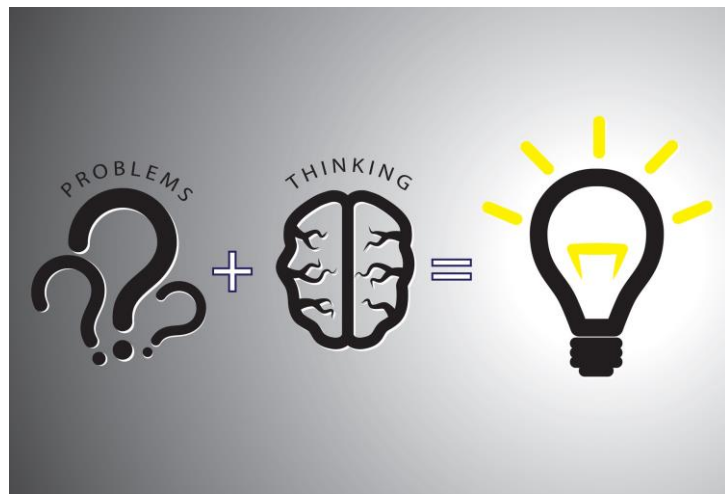
- 听说
 - “二分查找”
 - 用来查找给定目标
- 了解
 - 知道“二分查找”是“跳着找”
 - 因为“跳”，所以快
 - 因为“跳”，所以要求数据排序
- 理解
 - 知道每一个步骤及其细节
 - 会画算法流程图



深度掌握二分查找

- 实现
 - 用Python语言编写出无bug的二分查找算法
- 应用
 - 一个实际问题：请你只用加减乘除求取一个距离一个正整数的平方根最近的那个正整数
 - 你能应用“二分查找”算法，编程解决上面那个问题吗？





谢谢！