Lesson 01

伪代码
Pseudo Code

背景

问题：

给定一个数组 arr, 里面都是正整数, 而且没有重复

求里面最大的数

解决方案：

```java
public int max(int[] arr){
    int max = 0;
    for (int i = 0; i < arr.length; i++) {
        int value = arr[i];
        if (value > max) {
            max = value;
        }
    }
    return max;
}
```

分析：

代码写的很好，但是每个语言都可以用

什么是

一个人能读懂的代码, 跟具体实现语言无关

专业案例

$$
\begin{aligned}
&Max(A, n) \\
&\quad max \leftarrow 0 \\
&\quad \text{for } i \leftarrow 1 \text{ to } n \\
&\quad\quad \text{if } A_i > max \text{ then} \\
&\quad\quad\quad max \leftarrow A_i \\
&\quad \text{return } max
\end{aligned}
$$

伪代码与代码对比

省略声明：

省略变量声明, 省略函数返回值类型声明

数组：

arr vs A

索引 从 1 开始

可能不面向对象：

$$
\begin{aligned}
&Max(A, n)\\
&\quad max \leftarrow 0\\
&\quad \text{for } i \leftarrow 1 \text{ to } n\\
&\quad\quad \text{if } A_i > max \text{ then}\\
&\quad\quad\quad max \leftarrow A_i\\
&\quad \text{return } max
\end{aligned}
$$

vs

$$
\begin{aligned}
&Max(A)\\
&\quad max \leftarrow 0\\
&\quad \text{for } i \leftarrow 1 \text{ to } A.n\\
&\quad\quad \text{if } A_i > max \text{ then}\\
&\quad\quad\quad max \leftarrow A_i\\
&\quad \text{return } max
\end{aligned}
$$

算法
Algorithm

什么是

程序的思路！！！

解释：

针对一个特定问题的 程序解决步骤

跟语言无关

可以用文字、伪代码、代码表达，一般用伪代码表达

算法分类

基础算法：

1 CPU + 1 Memory

数据库算法：

1 CPU + 1 Memory + 1 Disk

内存不够 放到文件里存储

平行算法：

N CPU + 1 Memory @ 1 Computer

一台计算机 多核同时运算

分布式算法：

N Computer + Network

多台计算机 同时运算

AI 算法：

Data

基于数据分析 做出 预测

图像 算法：

Pixel

计算像素点

算法方法

设计算法的不同的思路

思路的思路

比如：

穷举法 / Brute Force

递归降级 / Recursion

分治 / Divide and Conquer

动态规划 / Dynamic Programming

贪心 / Greedy

互动

问题：

给定一个数组 A 里面都是正整数, 而且没有重复

求取两个数只和的最大值

时间复杂度
Time Complexity

时间消耗量化

干嘛?

要比较两个算法谁快谁慢

比较需要量化

量化案例

```java
public boolean exist(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        int value = arr[i];
        if (value == 0) {
            return true;
        }
    }
    return false;
}
```

思考:

有哪些会影响代码的时间?

影响代码时间的因素

代码

硬件

数据量

数据内容

数据内容分析

最差的情况 Worst Case

平均的情况 Average Case

最好的情况 Best Case

最差和平均情况分析有价值

时间消耗的量化基准

代码：

当前代码

硬件：

忽略硬件的影响 认为是公平竞争

数据量：

N

数据内容：

针对最差情况的数据内容

时间消耗计算

计算规则

以最小运算单元 为一个量，

对 n 个大小的输入数据求和

用 T(n) 表达

样例：

```java
public boolean exist(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        int value = arr[i];
        if (value == 0) {
            return true;
        }
    }
    return false;
}
```

```
for () { x n
    1 + 1
    1
}
n x (1 + 1 + 1)
3n
```

挑战

A:

```
public void run(int[] arr) {
    for (int i = 0; i < arr.length; i++){
        arr[i] = 0;
    }
}
```

B:

```
public void run(int[] arr) {
    for (int i = 1; i < arr.length; i *= 2){
        arr[i] = 0;
    }
}
```

C:

```
public void run(int[] arr) {
    for (int i = 0; i < arr.length; i++){
        for (int j = 0; j < arr.length; j++){
            arr[i] = arr[j];
        }
    }
}
```

D:

```
public void run(int[] arr) {
    for (int i = 0; i < arr.length; i++){
        for (int j = 0; j < i; j++){
            arr[i] = arr[j];
        }
    }
}
```

E:

```
public void run(int[] arr) {
    for (int i = 1; i < arr.length; i *= 2){
        for (int j = 0; j < i; j++){
            arr[i] = arr[j];
        }
    }
}
```

一些结论

两个 for 循环 不一定会慢

挑战

计算你的算法的时间消耗量化值