计算思维与实践

实验二 算法初步

TAKE YOU ON A TRIP
GO!

哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

探索 从未停止

目录

CONTENT

实验目的

- 实现无序数组和有序数组的基本操作(查找、插入、删除)。
- 通过性能对比，理解不同数据组织方式对操作效率的影响。
- 掌握时间复杂度分析，并能根据实际场景选择合适的数据结构

实验内容

为广告系统构建用户信息库（设计有序/无序数组）

一、利用无序数组来实现广告用户信息管理系统

- 设计一个无序数组来存储用户信息(包括用户id和兴趣分值)。
- 实现功能:
  1. 插入：将新用户信息直接插入数组末尾(O(1))
  2. 查找：按用户id线性查找，返回对应用户的兴趣分值(O(n))
  3. 删除：按用户id查找后删除，将被删除元素后面的所有元素依次向前移动一位(O(n))

实验内容

二、利用有序数组来实现广告用户信息管理系统

- 设计一个按用户id升序存储的数组
- 实现功能:
  1. 查找：按用户id二分查找，输出对应用户的兴趣分值(O(logn))
  2. 插入：按用户id找到插入位置，将插入位置及其后面的所有元素向后移动一位后插入新数据(O(n))
  3. 删除：按用户id二分查找后删除，并移动后续元素(O(n))

三、性能对比实验（不需提交）

1. 分别对无序数组和有序数组进行大量数据的插入、查找和删除操作，记录时间或操作次数。
2. 分析比较两种结构在不同操作上的性能差异。

实验内容

【输入样例】

```
UA_INSERT 101 9.8    //在无序数组中插入一个客户：ID = 101，兴趣分值 = 9.8
UA_INSERT 205 6.7    //插入客户：ID = 205，兴趣分值 = 6.7
UA_INSERT 123 8.5    //插入客户：ID = 123，兴趣分值 = 8.5 ->[101, 205, 123]
UA_FIND 205          //在无序数组中查找 ID = 205 的客户->FOUND 6.7
UA_DELETE 101        //删除 ID = 101 的客户
PRINT_UA             // 打印当前无序数组的所有客户信息->[205, 123]
OA_INSERT 105 7.3    //在有序数组中插入客户：ID = 105，兴趣分值 = 7.3
OA_INSERT 101 9.2    //插入客户：ID = 101，兴趣分值 = 9.2->[101, 105] 升序
OA_INSERT 109 6.8
OA_INSERT 106 8.5 ->[101, 105, 106, 109]
PRINT_OA             //打印当前有序数组
OA_FIND 106          //在有序数组中查找 ID = 106
OA_DELETE 105        //删除 ID = 105 的客户
PRINT_OA             //再次打印当前有序数组
END                  //结束程序
```

【输出样例】

```
UA_INSERT OK 1
UA_INSERT OK 2
UA_INSERT OK 3
UA_FIND FOUND 6.70
UA_DELETE OK 2
---- UNORDERED ARRAY ----
[00000] id=205 interest=6.70
[00001] id=123 interest=8.50
OA_INSERT OK 1
OA_INSERT OK 2
OA_INSERT OK 3
OA_INSERT OK 4
---- ORDERED ARRAY ----
[00000] id=101 interest=9.20
[00001] id=105 interest=7.30
[00002] id=106 interest=8.50
[00003] id=109 interest=6.80
OA_FIND FOUND 8.50
OA_DELETE OK 3
---- ORDERED ARRAY ----
[00000] id=101 interest=9.20
[00001] id=106 interest=8.50
[00002] id=109 interest=6.80
```

实验内容

客户结构体的定义：

```
#ifndef CUSTOMER_H
#define CUSTOMER_H
// 客户结构体：包含客户ID和兴趣分值（核心排序依据）
typedef struct {
    int id;             // 客户唯一标识
    float interest;     // 兴趣分值
} Customer;
#endif

```

实验内容

本实验要求补全 6 个函数，分别实现 **无序数组** 与 **有序数组** 的增、删、查操作。所有函数均以 Customer 数组为操作对象，并可选地记录性能指标（Metrics）：

一、无序数组部分（Unordered Array）

1.uaFindInterestById() – 线性查找

【功能】顺序遍历数组，根据 id 查找目标客户并返回其 interest（兴趣值）。
若未找到返回 INTEREST_NOT_FOUND。
每比较一次，可执行 if (m) m->compares++; 统计比较次数。

2.uaInsertBack() – 尾部插入

【功能】在数组尾部追加一个新客户。
若数组已满（n == capacity），返回 -1 表示插入失败；否则写入 customers[n] = c 并返回 n+1。
若统计启用，每次写入计一次 m->moves++。

3.uaDeleteById() – 按 id 删除

【功能】找到指定 id 的客户并删除，将后续元素顺次前移。
若没找到返回 -1，否则返回新长度 n-1。
每次比较或移动都可统计 m->compares++、m->moves++。

实验内容

二、有序数组部分（Ordered Array, 按 id 升序）

4.oaFindInterestById() – 二分查找

【功能】在升序数组中，用二分法查找指定 id，返回兴趣值或 INTEREST_NOT_FOUND。
每次比较计入 m->compares++。

5. oaInsertKeepOrder() – 保序插入

【功能】向有序数组中插入新客户，保持按 id 升序。
先二分定位插入位置，再将后续元素右移一格。
若数组已满返回 -1；否则插入并返回 n+1。
移动/比较操作均可计入统计。

6.oaDeleteById() – 有序数组删除

【功能】先用二分查找找到 id，再前移覆盖实现删除。
若未找到返回 -1；否则返回 n-1。

实验内容

无序数组参考模板：

```c
/* ================= 无序数组 ================= */

/* 线性查找（返回interest或INTEREST_NOT_FOUND） */
// 参数：customers-客户数组，n-数组长度，id-目标id，m-统计指针
float uaFindInterestById(const Customer customers[], int n, int id, Metrics *m) {
    // TODO:
    // 1) for i=0..n-1 线性扫描
    // 2) 每比较一次可执行 if(m) m->compares++;
    // 3) 若 customers[i].id == id 返回 customers[i].interest
    // 4) 未找到返回 INTEREST_NOT_FOUND
}

/* 尾部插入（成功返回新n，失败返回-1） */
// 参数：customers-客户数组，n-当前元素个数，capacity-最大容量，c-新客户，m-统计指针
int uaInsertBack(Customer customers[], int n, int capacity, Customer c, Metrics *m) {
    // TODO:
    // 1) 若 n == capacity → 返回 -1（容量不足）
    // 2) customers[n] = c; 若统计则 if(m) m->moves++;
    // 3) 返回 n+1
}

/* 按id删除（成功返回新n，未找到返回-1） */
// 参数：customers-客户数组，n-当前元素个数，id-目标id，m-统计指针
int uaDeleteById(Customer customers[], int n, int id, Metrics *m) {
    // TODO:
    // 1) 线性查找目标下标 idx（比较时可 if(m) m->compares++）
    // 2) 未找到 → 返回 -1
    // 3) for i=idx+1..n-1: customers[i-1] = customers[i]; 若统计则 moves++
    // 4) 返回 n-1
}
```

有序数组参考模板：

```c
/* ================= 有序数组（按id升序） ================= */

/* 二分查找（返回interest或INTEREST_NOT_FOUND） */
// 参数：customers-有序数组，n-数组长度，id-目标id，m-统计指针
float oaFindInterestById(const Customer customers[], int n, int id, Metrics *m) {
    // TODO:
    // 1) lo=0, hi=n-1
    // 2) while (lo<=hi): mid=(lo+hi)/2
    //    比较时可 if(m) m->compares++;
    //    a) customers[mid].id == id → 返回 customers[mid].interest
    //    b) customers[mid].id < id → lo=mid+1; 否则 hi=mid-1
    // 3) 未找到 → 返回 INTEREST_NOT_FOUND
}

/* 保序插入（成功返回新n，失败返回-1） */
// 参数：customers-有序数组，n-当前元素个数，capacity-最大容量，c-新客户，m-统计指针
int oaInsertKeepOrder(Customer customers[], int n, int capacity, Customer c, Metrics *m) {
    // TODO:
    // 1) 容量判定：若 n == capacity → 返回 -1
    // 2) 用二分“下界”定位插入位置 pos：第一个使 customers[pos].id >= c.id 的位置（0..n）
    //    比较过程中可 if(m) m->compares++;
    // 3) for i=n..pos+1 递减：customers[i] = customers[i-1]；若统计 moves++
    // 4) customers[pos] = c；若统计 moves++
    // 5) 返回 n+1
}

/* 按id删除（成功返回新n，未找到返回-1） */
// 参数：customers-有序数组，n-当前元素个数，id-目标id，m-统计指针
int oaDeleteById(Customer customers[], int n, int id, Metrics *m) {
    // TODO:
    // 1) 先二分查找 idx；比较时可 if(m) m->compares++;
    // 2) 未找到 → 返回 -1
    // 3) for i=idx+1..n-1: customers[i-1] = customers[i]；若统计 moves++
    // 4) 返回 n-1
}
```

课程平台使用指南

- 登录网址：http://10.249.41.7:9000/

- 常见oj错误提示：

① Compile Error　编译错误，不符合语法规范

② Wrong Output　表示运行结果不正确

③ Representation error　一般表示输出格式不正确

④ Runtime error　请检查是否有除零、数组越界、eles语句块没有用{}括起来等

⑤ Nonzero Exit Status　一般表示主函数返回非0值

调试方法1—打印语句

范例程序：
韩信点兵

今有物不知其数，
三三数之剩二，
五五数之剩三，
七七数之剩二。
问物几何?

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int x=1,find=0;
7
8      printf("before while\n");
9
10     while(!find);
11     {
12         if(x%3==2 && x%5==3 && x%7==2)
13         {
14             printf("x = %d\n",x);
15             find = 1;
16             x++;
17         }
18
19     printf("in while:x=%d\n",x);
20
21     }
22     return 0;
23 }
```

调试方法2—调试器

1. 开启调试器

中止运行

返回上一级

单步进入（step into）

运行一行（next line）

运行到光标 （run to cursor）

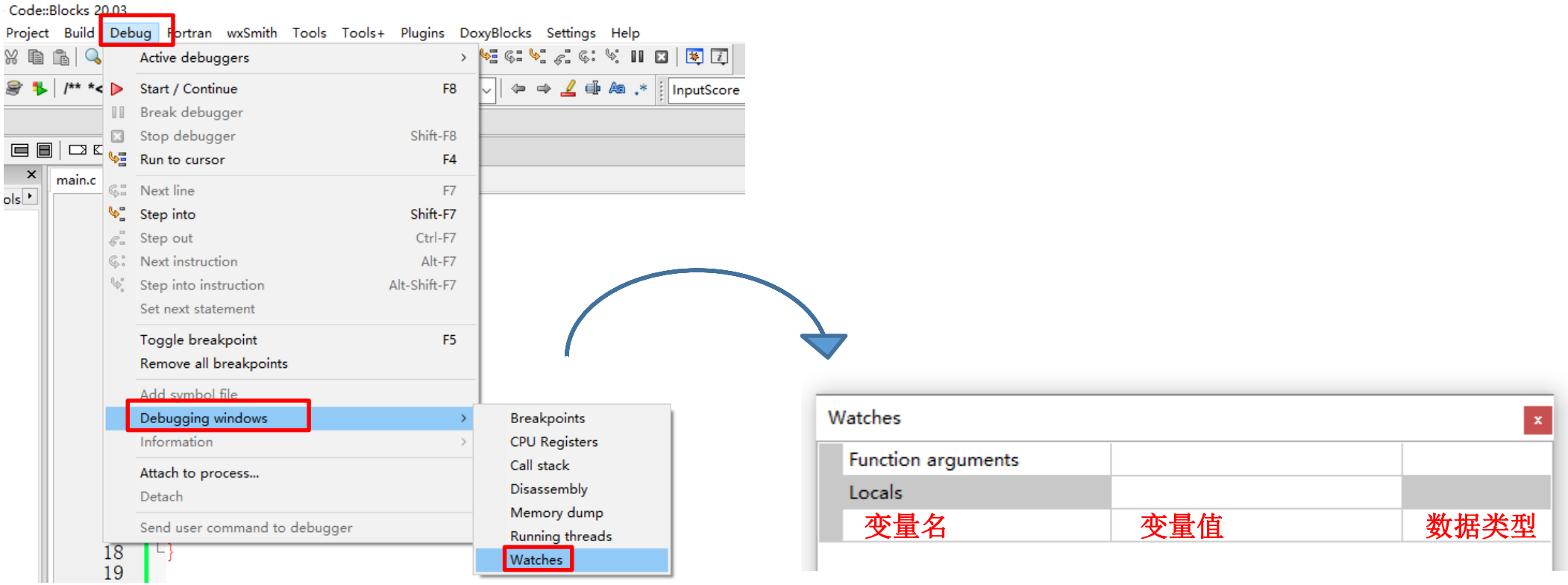
开始调试/运行到断点

断点：代码中设置的一个 “暂停标记”，当程序执行到这个标记位置时，会自动暂停运行，可以检查此时的变量值、程序执行流程等，从而精准定位错误。

注意：项目名称和保存路径不能有中文和空格，否则可能无法调试

调试方法2—调试器

2. 打开监视窗



- Function arguments: 函数参数的值
- Locals：当前有效的局部变量的值
- 空白行：可添加想要查看的变量，选中自定义变量所在的行，右键可删除特定行。

请同学们开始实验

GO!
TAKE YOU ON A TRIP

哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

探索 从未停止