**北京科技大学实验报告**

学院： 计通学院 专业：计算机科学与技术 班级： 计1503

姓名： 唐誉源 学号：41503302 实验日期： 2017 年 11 月 16 日

#### **实验名称：**

算法设计与分析实验3挑选玩具

#### **实验目的：**

1. 利用贪心算法完成玩具分组的问题
2. 保证玩具不重复且价值相差尽量小

#### **实验设备（环境）及要求：**

操作系统: macOS High Sierra 10.12.1;

编辑器: sublime txt 3;

编译链接: gcc.

#### **实验原理：**

贪心算法在对问题求解时，总是做出在当前看来是最好的选择，不从整体最优上加以考虑，她所做出的是在某种意义上的局部最优解。

在本题的应用场景下，利用贪心算法的思想，可以设定将最大的玩具依次分配给m组玩具中价值最小的玩具组。为了优化运算时间，选用冒泡排序的思想，快速得到最大的玩具。

在本题的情景下，贪心算法不是总能到最优解。

#### **实验内容与步骤：**

##### *（1）实验内容*

m人去超市挑选玩具，现有n个玩具，每个玩具的价值为，每个人挑选的玩具不能重复。m个人在挑选玩具时，都不希望其余的人多得，自己少得。所以需要把这些玩具分成m份，是他们的价值相差尽量小。

输入：测试数据占两行，第一行2个整数和，接下来一行，n个整数

输出：对每组测试数据，输出m份玩具价值相差的最小值，每组数据占一行

##### *（2）主要步骤*

由冒泡排序的思想得到最大的玩具，即依次选择玩具向后比较，如果比后边的玩具大，则进行交换，遍历一次之后，得到最后的那个玩具是价值最大的玩具。此时遍历m组玩具，选择价值最小的玩具组，将前述方法所得的价值最大的玩具分到这组。由此得到：

void swap(int \*a, int \*b) {  
 int temp = \*a;  
 \*a = \*b;  
 \*b = temp;  
}  
  
int split(int n, int m, int v[]) {  
 int g[m], i, j, k, d=0, flag;  
 **for** (i=0; i<m; i++){  
 g[i] = 0;  
 }  
 **for** (j=n-1; j>=0; j--) {  
 **for** (i=0; i<j; i++) {  
 **if** (v[i] > v[i+1]) {  
 swap(&v[i], &v[i+1]);  
 }  
 }  
 **for** (k=0, flag=0; k<m; k++) {  
 **if** (g[flag]>g[k]) {  
 flag = k;  
 }  
 }  
 g[flag] += v[j];  
 }  
 **for** (i=0; i<m-1; i++){  
 d += abs(g[i+1]-g[i]);  
 }  
 **return** d;  
}

在设计main函数时，读入测试数据，加入了输入数据范围的限制，如果超出范围则报错跳出。

int main() {  
 int n, m;  
 int v[1000];  
 printf("please input testing data (m<=n<=1000, 0<=vi<=1000):\n");  
 scanf("%d %d", &n, &m);  
 **if** (n>1000||m>n) {  
 printf("Wrong number. Please retry.\n");  
 **return** 1;  
 }  
 **for** (int i = 0; i < n; i++) {  
 scanf("%d", &v[i]);  
 }  
 printf("the smallest price difference is:%d\n", split(n, m, v));  
 **return** 0;  
}

#### **实验数据：**

样例输入：

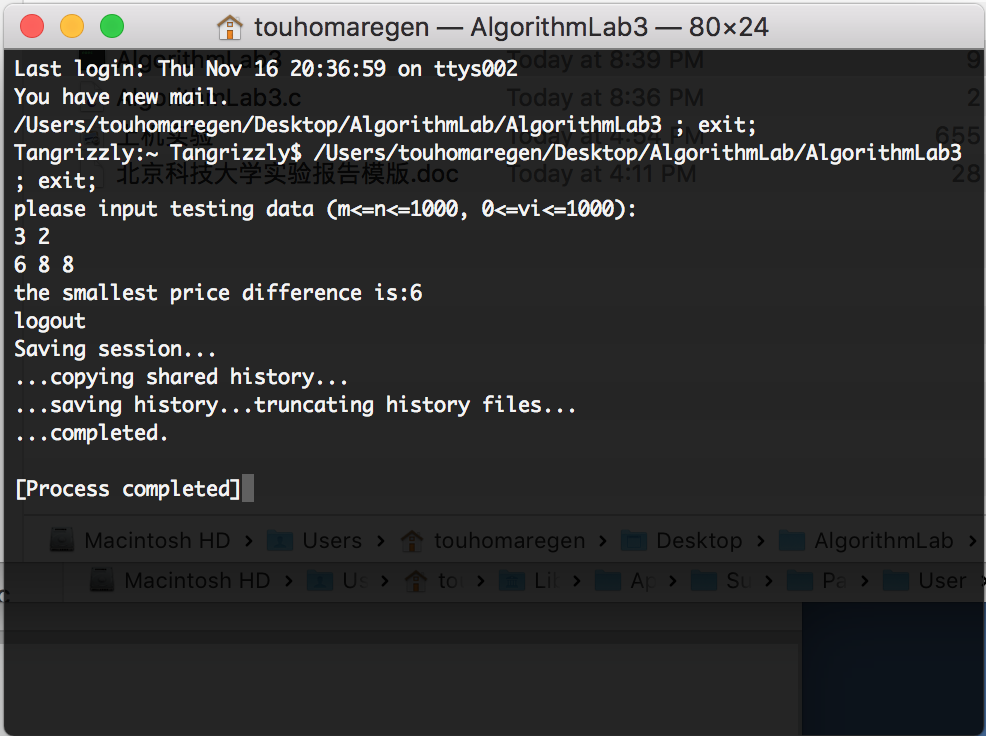
3 2  
6 8 8

样例输出：

6

#### **实验结果与分析：**

运行结果截图展示：



由图可视，3个玩具分给2个人，3个玩具的价值分别为6、8、8，分成价值差最小的两组为14和8，所得价值差为6。

#### **实验结论：**

##### *（1）实验结论*

熟悉了贪心算法设计程序的思想和方法，并利用她完成了玩具分组问题。逐个将玩具分组并保证在每一步达到最优解使得最终分组的玩具价值相差尽量小。在设计程序的过程中，保证了输入数据的有效性，增强了程序的健壮性。

##### *（2）讨论*

1. 在本题的场景下，是一个np问题，如果不遍历全部可能的情况，难以得到一个全局最优解，本题所用的贪心算法保证了在分每一个玩具时的局部最优解，但可能不是全局最优解；
2. 题目要求分组结果的价值差尽量小，但价值差的计算方法有很多种，在这里可能会造成歧义，在此所用的就是最简单的计算价值差的方法，依次用后一个组的价值减前一个组的价值；
3. 在排序时使用简单的排序算法思想，即冒泡排序法，其时间复杂度是，可以优化成更加复杂的排序算法如堆排序等减少时间复杂度；
4. 每个输入数值都设置了范围检测，如果不符合范围，则报错跳出，保证程序的在各种情况下的健壮性；
5. 程序由gcc直接编译得到Unix executable格式文件，在window上运行需要重新编译连接得到.exe文件。