关于排序算法的报告

作者：宁依华　　学号：1700015403

【摘要】目前我们已经在慕课课堂了解过三种排序算法，除了冒泡排序这样常规的排序法之外，也提到了 bogo 排序这种非常规的算法。目前的关于排序问题还有不少有趣的算法，本文旨在整理一些常规的和特殊的排序算法，尝试给出其代码，并对经典的排序问题的解决方法提出自己的认识。

【关键词】排序 算法 算法实现

选择排序

1、过程描述：

首先找到数组中最小的那个元素，将它和数组的第一个元素交换位置(如果第一个元素就是最小元素那么它就和自己交换)。接下来，在剩下的元素中找到最小的元素，将它与数组的第二个元素交换位置。如此往复，直到将整个数组排序。这种方法我们称之为选择排序。

2、代码实现：

```
public class SelectSort {
     public static int[] selectSort(int[] a) {
         int n = a.length;
        for (int i = 0; i < n - 1; i++) {
             int min = i;
             for (int j = i + 1; j < n; j++) {
                if(a[min] > a[j]) min = j;
            }
            //交换
            int temp = a[i];
           a[i] = a[min];
            a[min] = temp;
        }
        return a;
    }
}
```

3、时间复杂度[1]：

选择排序算法所需时间与 n2　成正比，记前时间复杂度为 O(n2)。我们熟知的冒泡排序时间复杂度也是 O(n2).

[1] 本文时间复杂度的引用来自百度

Bogo 排序

1、**过程描述：**

将所给的数随机打乱一次，如果打乱后正好排好，就结束程序，如果没有正好排好，就再打乱一次，直至排好。

2、代码实现：

```
import random

def bogosort(l):
    while not in_order(l):
        random.shuffle(l)
    return l

def in_order(l):
    if not l:
        return True
    last = l[0]
    for x in l[1:]:
        if x < last:
            return False
        last = x
    return True
```

3、时间复杂度：

猴子排序的时间复杂度为O(n*n!)。

睡眠排序

1、 过程描述

核心思想是有多少个待排序的数字，就开多少个协程，让每个线程睡对应的数字时长，沉睡之后打印对应数字，这样数字小的就会先打印了，数字大的就后打印了，实现了排序功能。

2、 优缺点分析

优点：最大程度地利用线程。

缺点：

1. 当出现一个很大的数字，会睡眠很长时间（效率低）
2. 当数字相差很小，会不精准（不精准）
3. 不能处理负数（虽然可以在加上一个正数，再进行睡眠排序，但是得不偿失）
4. 排序使用到了多线程，有点“杀鸡焉用宰牛刀”的感觉

面条排序

1. 过程描述：

面条排序的思想是：一组数字 array，拿来一把面条，array 里的最大值对应面条的高度，数字和面条高度由此可以得到一个系数，这个系数就是数字数值和面条高度的映射关系（比如，数字是 1 到 10，面条是十根，每一根面条都是高度 10，那这个系数就是 1），array 里的数字对应每一根面条，再把多出的高度给折掉，最终就得到长短不一的十根面条，然后把十根面条放桌子上端平，用手平行于桌面，从上往下，最先碰到手的，先拿开。这样面条排好序了，array 也排好序了。

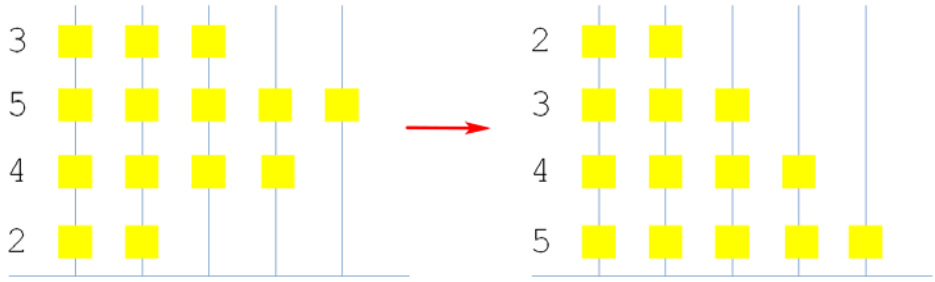
2、 代码描述

笔者不才，无法写出。有人说面条排序中，把面条放桌子上端平这个操作不能用代码实现，但笔者也看见网络上有人声称写出来面条排序的代码，为严谨起见，不在此处附上。

珠排序

1、过程描述

我们知道，在重力的作用下，如果将算盘立起来，无论怎么拨动，算珠最终都会掉下来紧挨在一起，那如果每根柱子上的算珠个数不太一样呢？就会出现下图的情况。



如图，我们看到神奇的重力已经帮我们排好序了。那如何将此现象写成代码呢？可以用一个二维数组模拟整个算盘，每一个数拥有一行算珠。

2、代码实现（系引用 JAVA 代码）

```
public void pearlSort(int[] arr) {
11          int min=arr[0];
12          int max=arr[0];
13          for (int i = 0; i < arr.length; i++) {
14              if(min>arr[i]) {
15                  min=arr[i];
16              }
17              if(max<arr[i]) {
18                  max=arr[i];
19              }
20          }
```

```
21
22          /*
23           * 定义二维数组
24           */
25          int[][] pal=new int[arr.length][max-min+1];
26          /*
27           * 给二维数组串上珠子
28           */
29          for (int i = 0; i < pal.length; i++) {
30              for (int j = 0; j < arr[i]-min; j++) {
31                  pal[i][j]=1;
32              }
33              pal[i][pal[i].length-1]=arr[i];
34          }
35
36          /*
37           * 珠子往下落
38           */
39          for (int i = 0; i < pal.length; i++) {
40              for (int v = pal.length-1-i; v > 0; v--) {
41                  for (int j = 0; j < pal[v].length-1; j++) {
42                      if(pal[v][j]!=0&&pal[v-1][j]==0) {
43                          pal[v][j]=0;
44                          pal[v][pal[v].length-1]--;
45                          pal[v-1][j]=1;
46                          pal[v-1][pal[v].length-1]++;
47                      }
48                  }
49              }
50              /*
51               * 依次将剩余的最大值赋给原数组
52               */
53              arr[arr.length-1-i]=pal[i][pal[i].length-1];
54          }
55      }
```

3、珠排序是一种模拟自然界规律的排序方式，但计算机运算复杂度相当高，远高于物理实现。

评价

其实程序员门发明的排序算法远不止这几种，在众多排序算法中，有一些容易想到且容

易实现的算法，诸如选择排序和冒泡排序，但也有像 boga 排序这样类似于恶搞、完全靠运气的算法，像珠排序这样从自然界获得灵感的算法，以及像睡眠排序这样优点缺点都很明显的算法。这些算法大多数看上去现在都是没有用武之地的，但或许将来就会出现适合他们的使用场景。

【参考文献】

1. 《P vs. NP：从一则数学家谋杀案说起》https://www.guokr.com/article/437662/
2. 选择排序-百度百科
3. 《十大经典排序算法》https://www.cnblogs.com/itsharehome/p/11058010.html
4. 排序算法——猴子排序（Bogosort）【代码实现】https://blog.csdn.net/qq_36721220/article/details/91351199
5. 神奇排序--睡眠排序 https://www.cnblogs.com/bigsaltfish/p/10067008.html