

## 《算法分析与设计》第2次作业

姓名：孙赫阳 学号：71118124

### 算法分析题

题目1：求下列递推关系式表示的算法复杂度。

$$(1) \quad T(n) = 9T\left(\frac{n}{3}\right) + n$$

$$(2) \quad T(n) = n + 3T\left(\frac{n}{4}\right)$$

$$(3) \quad T(n) = 4T\left(\frac{n}{2}\right) + n^2 \lg n$$

答：

由主定理 $T(n) = n^{\log_m k} + \sum_{j=0}^{\log_m n-1} k^j f(n/m^j)$ 可得：

$$(1) T(n) = n^{\log_3 9} + \sum_{j=0}^{\log_3 n-1} 9^j n/3^j = n^2 + n \sum_{j=0}^{\log_3 n-1} 3^j = n^2 + \frac{1}{6} n^2 -$$

$$\frac{1}{2} n = \frac{7}{6} n^2 - \frac{1}{2} n, \text{ 复杂度 } T(n) = \Theta(n^2).$$

$$(2) T(n) = n^{\log_4 3} + \sum_{j=0}^{\log_4 n-1} 3^j n/4^j = n^{\log_4 3} + n \sum_{j=0}^{\log_4 n-1} \left(\frac{3}{4}\right)^j = n^{\log_4 3} -$$

$$\frac{16}{3} n^{\log_4 3} + 4n = 4n - \frac{13}{3} n^{\log_4 3}, \text{ 复杂度 } T(n) = \Theta(n).$$

$$(3) T(n) = n^{\log_2 4} + \sum_{j=0}^{\log_2 n-1} 4^j n^2 \lg n / 2^j = n^2 + n^2 \lg n \sum_{j=0}^{\log_2 n-1} 2^j = n^2 +$$

$$\frac{1}{2} n^3 \lg n - n^2 \lg n, \text{ 复杂度 } T(n) = \Theta(n^3 \lg n) = \Theta(n^3 \log n).$$

题目2：假设谷歌公司在过去  $n$  天中的股票价格记录在数组  $A[1..n]$  中，我们希望能从中找出两天的价格，其价格增幅最大，即找到  $A[i]$  和  $A[j]$  ( $i \leq j$ )，使得  $M = A[j] - A[i]$  的值最大。请设计一个时间复杂度不超过  $O(n \lg n)$  的分治算法。

答：

初始时减数为  $a[0]$ ，然后算法不断记录比当前减数更小的减数；max 初始化

要求：1、分析题请用书面化语言给出详细分析过程；2、实现题请先写出算法思想，其次用伪代码描述，C++源码可以采用在线提交的方式，提交密码：seu711181，用户名最好统一使用学号-姓名的格式。

为 0，因为当  $p=q$  时， $a[p] - a[q] = 0$ ，故  $\max$  的值不可能为负数。因此将下标  $p$  初始化为 0， $q$  从下标 1 处开始向后扫描，并计算  $\text{sub} = a[q] - a[p]$ ，若  $\text{sub}$  大于  $\max$ ，则更新  $\max$  和最大值序号  $\text{maxKey}$  的值。若  $\text{sub}$  小于 0，意味着找到一个新的数组元素，该数组元素比  $\text{arr}[p]$  的值要小，则更新下标  $p$ 。

当计算  $\text{sub} = a[q] - a[p]$  时， $a[p]$  越小，则得到的  $\text{sub}$  越大。而下标  $p$  不断标记更小的减数，后面的元素  $a[q]$  与更小的减数相减才能得到更大的差值。

由代码可知，函数体中只有一层循环，进行一次线性扫描即可得到结果，其余操作均为  $O(1)$ 。综上所述，该算法的复杂度为  $O(n)$ ，不超  $O(n \log n)$ ，符合要求。

*//本代码cpp 亦同附件上传*

`#include<iostream>`

`using namespace std;`

*//算法复杂度O(n)*

`void FindMaxGap(double a[],int n,int &i,int &j) {`

`double max = 0;`

`int p = 0, q, maxKey;`

`double sub;`

`for (q = 1; q < n; q++) {`

`sub = a[q] - a[p];`

`if (sub > max) {`

`max = sub;`

`maxKey = q;`

`}`

`else if (sub < 0) //此时a[q]比a[p]更小`

`p = q;`

`}`

`i = p;`

`j = maxKey;`

`}`

`int main() {`

`double A[5] = { 998.5,766.5,232.1,322.5,666.6 };`

`int i, j;`

`FindMaxGap(A, 5, i, j);`

`cout << i + 1 << ' ' << j + 1; //所求序号=数组序号+1`

`return 0;`

`}`

要求：1、分析题请用书面化语言给出详细分析过程；2、实现题请先写出算法思想，其次用伪代码描述，C++源码可以采用在线提交的方式，提交密码：seu711181，用户名最好统一使用学号-姓名的格式。

## 算法实现题

**题目 3：**问题描述：在与联盟的战斗中连续失败后，帝国撤退到最后一个据点。根据其强大的防御系统，帝国击退了联盟攻击的六波浪潮。经过几个不眠之夜，联盟将军亚瑟注意到防御系统的唯一弱点就是能源供应。该系统由  $N$  个核电站供电，其中任何一个都会使系统失效。

这位将军很快就派  $N$  名特工进行突击，这些特工进入了据点。然而，由于帝国空军的袭击，他们未能降落在预期位置。作为一名经验丰富的将军，亚瑟很快意识到他需要重新安排计划。他现在要知道的第一件事是哪个特工离任何一个核电站最近。你是否可以帮助将军计算特工与核电站之间的最小距离？

题目细节及提交地址：<https://vjudge.net/contest/359940>。

答：

**算法思想：**

该问题类似于课程中的最接近点对问题，只不过坐标系中的点要分为核电站和特工两个点集，同一点集中的点无需比较距离，或者说同一点集中的两点求距结果应为无穷大，以避免影响正确结果的计算。

对于输入的  $N$ （核电站）+ $N$ （特工）= $2N$  个点，以横坐标中位数为划分，将其分为大小为  $N$  两个子集，依次类推，分治求解最小距离，再加上对于划分边界处点的考虑，即可得到整个平面坐标系上特工与核电站之间的最小距离。

需要强调的是，通过先预先将这些点根据  $y$  坐标排好序的预处理方法，以及使用课堂上讲述的边界点只需在  $2dm \times dm$  的区域内检查 6 个点的方法，可以降低时间复杂度。

**$MinDistance(s, n)$**

1.  **$if\ n = 1$**

**$then\ d \leftarrow$  the distance between the only station and the only agent**

**and return  $d$**

2.  **$m \leftarrow$  the median number of  $n$  integers by points'  $x$  position in  $S$**

3.  **$s1 \leftarrow \{p \in s \mid x_p < m\}$  and  $s2 \leftarrow \{p \in s \mid x_p \geq m\}$**

4.  **$d1 \leftarrow MinDistance(s1, \frac{n}{2})$  and  $d2 \leftarrow MinDistance(s2, \frac{n}{2})$**

5.  **$d_{min} \leftarrow \min\{d_1, d_2\}$**

要求：1、分析题请用书面化语言给出详细分析过程；2、实现题请先写出算法思想，其次用伪代码描述，C++源码可以采用在线提交的方式，提交密码：seu711181，用户名最好统一使用学号-姓名的格式。

6.  $P1 \leftarrow \{p \mid p \in s1 \text{ and } |x_p - m| < d_{min}\}$

**and**  $P2 \leftarrow \{p \mid p \in s2 \text{ and } |x_p - m| < d_{min}\}$

7. **do** sort( $P1$ ) **and** sort( $P2$ ) according to points' y position

8.  $d3 \leftarrow$  the smallest distance found through scanning  $P1$  and  $P2$ ,

for each point in  $P1$ , check up to 6 points in  $P2$

9. **return**  $\min\{d_{min}, d3\}$

**时间复杂度分析：**

第 1 步仅在输入  $n=1$  时执行 1 次；

第 2 步寻找中位数可在  $O(n)$  内完成，且经过预处理排序后可以在  $O(1)$  常数时间完成；

第 3 步需要  $O(n)$  时间完成；

第 4 步需要  $2T(n/2)$ ；

第 5 步  $O(1)$ ；

第 6 步线性扫描一次即可完成，需要  $O(n)$ ；

第 7 步由于进行了预排序，所以在分治法中只需要一次线性扫描就可以抽取所需要的排好序的点列  $P1$  和  $P2$ ，然后在第 8 步中再对  $P1$  进行一次线性扫描，因此第 7 步和第 8 步合在一起需要  $O(n)$  时间；

第 9 步  $O(1)$ 。

综上所述  $T(n) = 2T(n/2) + O(n)$ ,  $n \rightarrow \infty$ ，由主定理得  $T(n) = O(n \log n)$ 。