线性时间选择算法时间复杂度深入研究

王云鹏
(厦门大学软件学院，厦门 361005)

摘 要： 算法研究是计算机科学的核心领域之一。文中针对元素选择问题及解此问题的线性时间选择算法进行了深入研究，详细分析并论证了期望情况下与最坏情况下线性时间选择算法的时间复杂度，并对拟中位数元素选择问题进行了深层次的拓展，通过计算比较求出了线性时间下的最小复杂度因子。以期有助于该算法在相关领域的应用。
关键词： 算法；时间复杂度；拟中位数；复杂度因子

In-depth Study to the Time Complexity of Linear Time Selection Algorithm

WANG Yunpeng
(Software School of Xiamen University, Xiamen 361005)

Abstract: The algorithm study is one of core fields in computer science.The paper makes a further study of element selection problem and the linear time selection algorithm which is used to solve the problem above.Moreover, it analyzes the time complexity of linear time selection algorithm respectively in the expected and worst circumstance in detail and simultaneously gives it a strict proof .At last,The paper extends the method of analogical median deeply in solving the element selection problem.After caculation and comparison,the smallest complexity factor with linear time has been found out.Hope it be helpful to the application in the relevant fields.
Key words: algorithm; time complexity; analogical median; complexity factor

1 引言

给定 n 个元素的集合，集合中的第 k 个顺序统计量是指集合中的第 k 个 ($1 \leqslant k \leqslant n$) 最小元素。集合中第一个顺序统计量是指集合中的最小元素，第 n 个顺序统计量是指集合中的最大元素。如何从给定的集合中查找第 k 个最小元素，被称为元素选择问题。解决这一问题的算法在数据文件分类、人工智能搜索以及并行算法和图论算法设计方面都具有广泛应用。当前，已有多种算法可以轻易地进行元素选择。不过，如何进一步降低线性时间选择算法的时间复杂度，以此提高大容量数据检索和人工智能搜索的速度与效率，一直以来都是比较排序算法领域研究的热点。

2 期望情况下线性时间的选择

下面讨论解一般选择问题的一个分治算法 RandomizedSelect。其基本思想是对输入数组进行递归划分，该划分由 RandomizedPartition 函数随机产生。因此，要找数组 a [0 : n-1] 中第 k 小元素，只要调用 RandomizedSelect (a,0,n-1,k) 即可。具体算法描述如下：

```
template<class Type>
Type RandomizedSelect (Type a [] , int p, int r, int k)
{
  if (p==r) return a [p] ;
  int i=RandomizedPartition (a, p, r) , j=i-p+1;
  if (k<=j) return RandomizedSelect (a, p, i, k) ;
  return RandomizedSelect (a, i+1, r, k-j) ;
}
```

基于以上算法，进行如下时间复杂度分析：

设 T (n) 表示 RandomizedSelect 算法在输入为 a [p..r] 时的运行时间。因为该算法为一个随机算法，所以 T (n) 是一个随机变量。过程 RandomizedPartition 以等概率返回任一元素作为枢轴元素，因此，对于满足 $1 \leqslant j \leqslant n$ 的每个 j，子数组 a [p..i] 有 j 个元素 (都小于等于枢轴元素) 概率为 1/n。对于 j=1,2 ……,n，定义指示器随机变量为 X_j

$$X_j = \{\text{子数组 } a\ [p..r] \text{ 只有 j 个元素}\} \text{ 且 } E\ [X_j] = \frac{1}{n}$$

假设 $T\ (n)$ 单调递增，分析在最大可能输入的情况下递归调用所用的时间，即得到一个上界。第 k 个元素总是在具有最大个数的那个划分中。对于给定的调用 RandomizedSelect，指示器随机变量 X_j 在只有一个 j 值时为 1；其他所有 j 值时则为 0。当 $X_j=1$ 时，设递归调用的两个子数组大小分别为 $j-1$ 和 $n-j$。因此，递归方程为

$$T(n) \leq \sum_{j=1}^{n} X_j \cdot (\mathrm{T}(\max(j-1, n-j)) + O(n))$$
$$= \sum_{j=1}^{n} (X_j \cdot T(\max(j-1, n-j)) + O(n))$$

两边同时取期望，则得

$$E[T(n)] \leq E[\sum_{j=1}^{n} (X_j T(\max(j-1, n-j)) + O(n)]$$
$$= \sum_{j=1}^{n} E[X_j \cdot T(\max(j-1, n-j))] + O(n)$$
$$= \sum_{j=1}^{n} E[X_j] \cdot E[T(\max(j-1, n-j))] + O(n)$$
$$= \sum_{j=1}^{n} \frac{1}{n} \cdot E[T(\max(j-1, n-j))] + O(n)$$

本文收稿日期：2009-3-30

其中 X_j 和 $T(\max(j-1,n-j))$ 是独立的随机变量。

考虑表达式 $\max(j-1,n-j)$，则有

$$\max(j-1,n-j)=\begin{cases} j-1 \Leftarrow j>\lceil n/2\rceil \\ n-j \Leftarrow j\le\lceil n/2\rceil \end{cases}$$

如果 n 为偶数，在求和算式中，从 $T(\lceil n/2\rceil)$ 到 $T(n-1)$ 中的每一项只出现两次；如果 n 为奇数，所有这些项出现两次，$T(\lfloor n/2\rfloor)$ 出现一次。因此，

$$E[T(n)]\le\frac{n}{2}\cdot\sum_{j=\lfloor n/2\rfloor}^{n-1}E[T(j)]+O(n)$$

用替换方法解此方程。假设对于满足递归方程初始条件的某些常数，c，$T(n)\le cn$ 且当 $n\le n_0$ 时，$T(n)=O(1)$，其中 n_0 为足够小的整数。设 $O(n)$ 项隐含的常数为 a，使得对于所有 $n>0, O(n)\le a\cdot n$。利用归纳假设

$$\begin{aligned}
E[T(n)] &\le \frac{2}{n}\cdot\sum_{j=\lceil n/2\rceil}^{n-1}cj+an \\
&=\frac{2c}{n}\left(\sum_{j=1}^{n-1}j-\sum_{j=1}^{\lfloor n/2\rfloor-1}j\right)+an \\
&=\frac{2c}{n}\left(\frac{(n-1)n}{2}-\frac{(\lfloor n/2\rfloor-1)\lfloor n/2\rfloor}{2}\right)+an \\
&\le\frac{2c}{n}\left(\frac{(n-1)n}{2}-\frac{(n/2-2)(n/2-1)}{2}\right)+an \\
&=\frac{2c}{n}\left(\frac{n^2-n}{2}-\frac{n^2/4-3n/2+2}{2}\right)+an \\
&=\frac{c}{n}\left(\frac{3n^2}{4}+\frac{n}{2}-2\right)+an \\
&=c\left(\frac{3n}{4}+\frac{1}{2}-\frac{2}{n}\right)+an \\
&\le\frac{3cn}{4}+\frac{c}{2}+an \\
&=cn-\left(\frac{cn}{4}-\frac{c}{2}-an\right)
\end{aligned}$$

需要证明，对于足够大的 n，最后的表达式至多为 cn，或 $\frac{cn}{4}-\frac{c}{2}-an\ge 0$。该不等式两边加上 $\frac{c}{2}$ 且提出公因子 n，可得 $n(\frac{c}{4}-a)\ge\frac{c}{2}$。只要选择常数 c 满足 $\frac{c}{4}-a>0$，即 $c>4a$，即 $c>4a$ 得

$$n\ge\frac{c/2}{c/4-a}=\frac{2c}{c-4a}$$

因此，选择 $c>4a$，$n_0=2c/(c-4a)$。假设当 $n\le n_0$ 时，$T(n)=O(1)$，可得 $T(n)=O(n)$，综合以上分析，得出一般选择问题平均情况下线性时间内即可完成。

3 最坏情况下线性时间的选择

如果能在线性时间内找到一个划分基准，使得按这个基准所划分出的 2 个子数组的长度都至少为原数组长度的 ε 倍 $(0<\varepsilon<1)$，那么就可以在最坏情况下用 $O(n)$ 时间完成选择任务。

可以按以下步骤找到满足要求的划分基准：

将 n 个输入元素划分成 én/5ù 个组，每组 5 个元素，只可能有一个组不是 5 个元素。用任意一种排序算法，将每组中的元素排好序，并取出每组的中位数，共 én/5ù 个。

递归调用 select 来找出这 én/5ù 个元素的中位数。如果 én/5ù 是偶数，就找它的 2 个中位数中较大的一个。以这个元素作为划分基准。

图 1 是上述划分策略的示意图，其中 n 个元素用小圆点来表示，空心小圆点为每组元素的中位数。中位数的中位数在图中标出。图中所画箭头是由较大元素指向较小元素的。为简化问题，设所有元素互不相同。在这种情况下，找出的基准 x 至少比 $3\lfloor(n-5)/10\rfloor$ 个元素大，因为在每一组中有 2 个元素小于本组的中位数，而 $\lfloor n/5\rfloor$ 个中位数中又有 $\lfloor(n-5)/10\rfloor$ 个小于基准 x。同理，基准 x 也至少比 $3\lfloor(n-5)/10\rfloor$ 个元素小。而当时 $n\ge 75$，$3\lfloor(n-5)/10\rfloor\ge n/4$。所以按此基准划分所得的 2 个子数组的长度都至少缩短 $\frac{1}{4}$。

具体算法描述如下：

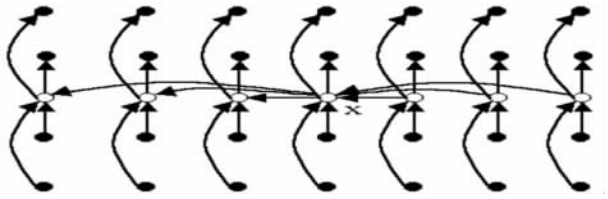

图 1 拟中位数法

```
Template < class Type >
Type Select (Type a [ ] , int p, int r, int k)    {
     if  (r–p<75)    {
用某个简单排序算法对数组 a [p:r] 排序;
  return a [p+k–1] ;
  } ;
for  ( int i = 0; i<= (r–p–4) /5; i++ )
将 a [p+5*i] 至 a [p+5*i+4] 的第 3 小元素与 a [p+i] 交换位置;
//找中位数的中位数，r–p–4 即上面所说的 n–5
Type x = Select (a, p, p+ (r–p–4) /5,  (r–p–4) /10) ;
int i=Partition (a,p,r, x) ,
j=i–p+1;
if  (k<=j)  return Select (a,p,i,k) ;
else return Select (a,i+1,r,k–j) ;
}
```

基于以上算法，进行如下时间复杂度分析。

设对 n 个元素的数组调用 Select 需要 $T(i)$ 时间，那么找中位数的中位数 x 至多用 $T(n/5)$ 时间。由于按照算法所选的基准 x 进行划分所得的两个子数组分别至多有 $\frac{3n}{4}$ 个元素，所以无论对哪一个子数组调用 Select 至多用 $T(3n/4)$ 时间。

（下转到 37 页）

并在这个项目中使学习者对 Session 这个作用域存取信息有了深入了解，这给学习者学习后续内容增添了信心，也带来了学习的乐趣。在这个项目中，学习者对 MVC 模式也能有一个很好的感性认识。

Session 对象用来保存每个用户的信息，以便跟踪每个用户的操作状态。其中，Session 的信息保存在容器中，而 SessionID 保存在客户机的 Cookie 中。讨论的购物车是基于 Cookie 技术的 Session，如果客户端关闭了 Cookie 功能，那么通常情况下，服务器会自动切换到 URL 重写来实现 Session 功能。

参考文献

[1] 张莉. 电子商务中购物车的实现. 福建电脑，2005: 128–129.

[2] 袁玉苹. 网上书店---购物车的设计与实现. 商场现代化，2008：145.

[3] 马莹. 电子购物车及实现技术. 绍兴文理学院学报，2002，22.

[4] 余正涛，宋丽哲，车文刚，郭剑毅. 网上购物车的数据库技术实现策略. 计算机应用. 2000，20 (8) .

作者简介

谌桂枝，女 (1970-)，株洲职业技术学院信息工程系计算机应用技术专业负责人，在职研究生，主要从事 J2EE 软件方向的研究。

(上接第 4 页)

$$T(n)=\begin{cases}C_1 \Leftarrow n<75\\ C_2 n+T(n/5)+T(3n/4) \Leftarrow n\geq 75\end{cases}$$

T (i) 的递归式中两个自变量之和 $n/5+3n/4=19n/20=an$, $0<a<1$。

这是使得 $T(n)=O(n)$ 的关键。

4 线性时间下的复杂度因子

采用拟中位数作为划分基准可以保证在最坏情况下用线性时间完成选择，$T(n)=O(n)$ 在量级上达到了该问题的下界。不过，子序列长度选择 5，是否同时获得了最小的时间复杂度呢？

设 Select 算法在最坏情况下时间复杂度为 $T(n)$，设算法中子序列长度选取 x，那么对每个子序列中的元素排序都需 $[\log(x!)]$ 次比较，总共则需要 $[\log(x!)]\cdot\lceil n/x\rceil$ 次比较，此时的 $T(n)$ 满足

$$T(n)=T(n/x)+T(3n/4)+(\lceil\log(x!)/x\rceil+1)n$$

因 $T(n)$ 为线性的，要求 $\frac{1}{x}+\frac{3}{4}<1$ 即 $x>4$，设 β 为复杂度因子，$T(n)=\beta\cdot n$，那么

$$\beta\cdot n=\beta n/x+\beta\cdot 3n/4+(\lceil\log(x!)/x\rceil+1)n$$

得，$\beta=4(x+\lceil\log(x!)\rceil)/(x-4)$

若令 $f(x)=(x+\lceil\log(x!)\rceil)/(x-4)$，当 $x>4$ 时，只需求出函数 f (x) 的最小值即可得最小复杂因子 β。考虑到算法中子序列长度的对称性，这里选取 x>4 的奇数。经计算得到如表 1 所示数据。

显然，子序列长度为 5 并非我们的最优选择。不难发现，当为子序列长度选取 19 时可以得到更小的复杂度因子=20.28。通过对拟中位数法解元素选择问题的深层次拓展，可以求出线性时间下的最小复杂度因子，那么线性时间选择算法最小的时间复杂度问题便迎刃而解。

表 1 子序列长度选取与复杂因子对应关系

x	5	9	13	17	19	21	25	29
$f(x)$	12	5.6	5.11	5.08	5.07	5.12	5.19	5.28
β	48	22.4	20.44	20.32	20.28	20.48	20.76	21.12

5 结语

在过去的半个世纪里，算法研究取得了大量重要的突破。同时，作为算法特性之一的时间复杂度也在算法爱好者的不懈探索中一降再降，这为从数据库系统到 Internet 搜索引擎，从 N 体问题的模拟到分子生物学中的序列分析等等一系列算法应用领域带来了革命。文中则通过对元素选择问题的深入研究，详细分析了期望情况下与最坏情况下的线性时间选择算法，并针对两种情况下算法的时间复杂度给出了严格的论证。最后，通过计算与比较得出线性时间下元素选择问题的最小复杂度因子。至此，对元素选择问题以及解决此问题的线性时间选择算法都有了更进一步的研究、更深层次的理解，在相关领域会得到广泛应用。

参考文献

[1] 王晓东. 计算机算法设计与分析第 3 版. 电子工业出版社，2007.

[2] 贺红，马绍汉. 算法分析与设计技术. 科学出版社，2004.

[3] Blum M, Floyd R W, Pratt V R,et al. Time Bounds for Selection, J.Computer and System Sciences, 1973, 7 (4) : 448–461.

[4] 钟 诚. VALIANT 并行归并及排序时间复杂性的分析研究. 广西大学学报 (自然科学版)，1997, 22 (4): 285–288.

作者简介

王云鹏，男 (1989-)，大三本科在读，主要研究方向：算法、ACM 编程。