数据结构与算法（八）

张铭 主讲

采用教材：张铭，王腾蛟，赵海燕 编写
高等教育出版社，2008. 6（“十一五”国家级规划教材）

http://www.jpk.pku.edu.cn/pkujpk/course/sjjg

大 纲

- 8.1 排序问题的基本概念
- 8.2 插入排序 (Shell 排序)
- 8.3 选择排序 (堆排序)
- 8.4 交换排序
  - 8.4.1 冒泡排序
  - 8.4.2 快速排序
- 8.5 归并排序
- 8.6 分配排序和索引排序
- **8.7 排序算法的时间代价**
- 内排序知识点总结

8.7 排序算法的时间代价

- 简单排序算法的时间代价
- 排序算法的理论和实验时间
- 排序问题的下限

原 因

- 一个长度为 n 的序列平均有 n(n-1)/4 对逆置
- 任何一种只对相邻记录进行比较的排序算法的平均时间代价都是 $\Theta(n^2)$

8.7.2 排序算法的理论和实验时间

算法	最大时间	平均时间	最小时间	辅助空间代价	稳定性
直接插入排序	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n)$	$\Theta(1)$	稳定
冒泡排序	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n)$	$\Theta(1)$	稳定
选择排序	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(1)$	不稳定

8.7.2 排序算法的理论和实验时间

算法	最大时间	平均时间	最小时间	辅助空间	稳定性
Shell排序(3)	$\Theta(n^{3/2})$	$\Theta(n^{3/2})$	$\Theta(n^{3/2})$	$\Theta(1)$	不稳定
快速排序	$\Theta(n^2)$	$\Theta(n\log n)$	$\Theta(n\log n)$	$\Theta(\log n)$	不稳定
归并排序	$\Theta(n\log n)$	$\Theta(n\log n)$	$\Theta(n\log n)$	$\Theta(n)$	稳定
堆排序	$\Theta(n\log n)$	$\Theta(n\log n)$	$\Theta(n\log n)$	$\Theta(1)$	不稳定
桶式排序	$\Theta(n+m)$	$\Theta(n+m)$	$\Theta(n+m)$	$\Theta(n+m)$	稳定
基数排序	$\Theta(d\ (n+r))$	$\Theta(d\ (n+r))$	$\Theta(d\ (n+r))$	$\Theta(n+r)$	稳定

小 结

- n 很小或基本有序时插入排序比较有效
- Shell 排序选择增量以3的倍数递减
  - 需要保证最后一趟增量为1
- 综合性能快速排序最佳

测试环境

- 硬件环境
  - CPU：Intel P4 3G
  - 内存：1G
- 软件环境
  - Windows XP
  - Visual C++ 6.0

随机生成待排序数组

```
//设置随机种子
inline void Randomize()   {
    srand(1);
}
// 返回一个[0,n-1]之间的随机整数值
inline int Random(int n) {
    return rand() % (n);
}
//产生随机数组
ELEM *sortarray =new ELEM[1000000];
for(int i=0; i<1000000; i++)
    sortarray[i] = Random(32003);
```

时间测试

```
#include <time.h>
#define CLOCKS_PER_SEC  1000
clock_t tstart = 0;  // 开始的时间
// 初始化计时器
void Settime()  {
    tstart = clock();
}
// 上次 Settime() 之后经过的时间
double Gettime() {
    return (double)((double)clock() -
        (double)tstart) / (double)CLOCKS_PER_SEC;
}
```

排序的时间测试

```
Settime();
for (i=0; i<ARRAYSIZE; i+=listsize) {
   sort<int>(&array[i], listsize);
}
cout << "Sort with list size " << listsize
<< ", array size " << ARRAYSIZE  << ", and
threshold " <<
THRESHOLD << ": "    << Gettime() << "
seconds\n";
```

8.7.2 排序算法的理论和实验时间

数组规模	10	100	1K	10K	100K	1M	10K 正序	10K 逆序
直接插入排序	0.00000047	0.000020	0.001782	0.1752	17.917	-----	0.00011	0.35094
选择排序	0.00000110	0.000041	0.002922	0.2778	36.500	-----	0.27781	0.29109
冒泡排序	0.00000160	0.000156	0.015620	1.5617	207.69	-----	0.00006	2.44840
Shell排序(2)	0.00000156	0.000036	0.000640	0.0109	0.1907	3.0579	0.00156	0.00312
Shell排序(3)	0.00000078	0.000016	0.000281	0.0038	0.0579	0.8204	0.00125	0.00687
堆排序	0.00000204	0.000027	0.000344	0.0042	0.0532	0.6891	0.00406	0.00375
快速排序	0.00000169	0.000021	0.000266	0.0030	0.0375	0.4782	0.00190	0.00199
优化快排/16	0.00000172	0.000020	0.000265	0.0020	0.0235	0.3610	0.00082	0.00088
优化快排/28	0.00000062	0.000011	0.000141	0.0018	0.0235	0.2594	0.00063	0.00063
归并排序	0.00000219	0.000028	0.000375	0.0045	0.0532	0.5969	0.00364	0.00360

8.7.2 排序算法的理论和实验时间

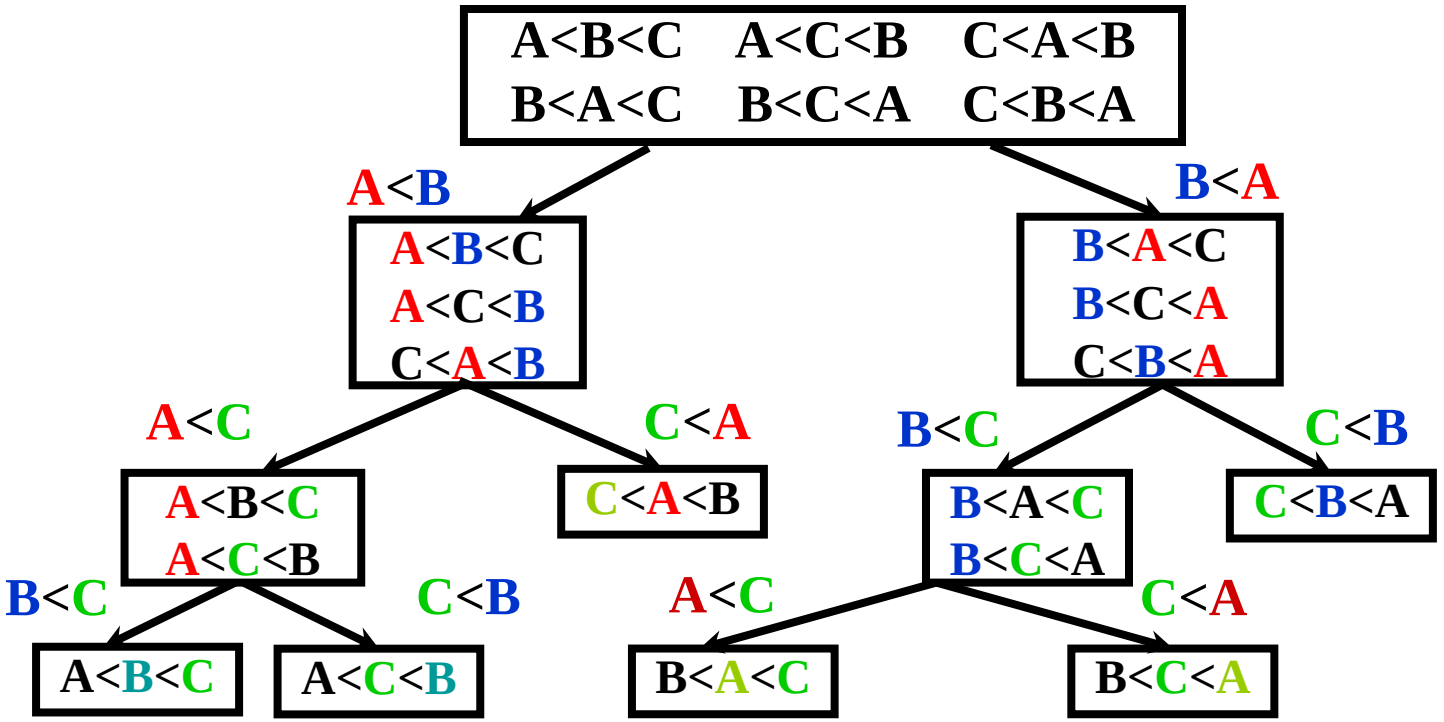
数组规模	10	100	1K	10K	100K	1M	10K 正序	10K 逆序
优化归并/16	0.00000063	0.000014	0.000188	0.0030	0.0375	0.4157	0.00203	0.00265
优化归并/28	0.00000062	0.000013	0.000204	0.0027	0.0360	0.4156	0.00172	0.00265
顺序基数/8	0.00000610	0.000049	0.000469	0.0048	0.0481	0.4813	0.00484	0.00469
顺序基数/16	0.00000485	0.000034	0.000329	0.0032	0.0324	0.3266	0.00328	0.00313
链式基数/2	0.00002578	0.000233	0.002297	0.0234	0.2409	3.4844	0.02246	0.02281
链式基数/4	0.00000922	0.000075	0.000719	0.0075	0.0773	1.3750	0.00719	0.00719
链式基数/8	0.00000704	0.000048	0.000466	0.0049	0.0502	0.9953	0.00469	0.00469
链式基数/16	0.00000516	0.000030	0.000266	0.0028	0.0295	0.6570	0.00281	0.00281
链式基数/32	0.00000500	0.000027	0.000235	0.0028	0.0297	0.5406	0.00263	0.00266

排序问题的下限

- 排序问题的时间代价在 $\Omega(n)$ (单趟扫描) 到 $O(n \log n)$ (平均, 最差情况) 之间
- 基于比较的排序算法的下限也为$\Omega(n \log n)$
- 用 **判定树 (Decision Tree)** 可以说明
  - 判定树中叶结点的最大深度就是排序算法在最差情况下需要的最大比较次数
  - 叶结点的最小深度就是最佳情况下的最小比较次数

8.7.3 排序问题的下限

用判定树模拟基于比较的排序



基于比较的排序的下限

- 对 n 个记录，共有 n! 个叶结点
- 判定树的深度至少为 $\log(n!)$
- 在判定树深度最小的情况下最多需要 $\log(n!)$ 次比较，即 $\Omega(n \cdot \log n)$
- 在最差情况下任何基于比较的排序算法都至少需要 $\Omega(n\log\ n)$ 次比较

排序问题的时间代价

- 排序问题需要的运行时间也就是

  $\Omega(n \cdot \log\ n)$

- 所有排序算法都需要 $O(n \cdot \log\ n)$的运行时间，因此可以推导出排序问题的时间代价为 $\Theta\ (n \cdot \log\ n)$

基数排序效率

- 时间代价为 $\theta(d\ (n+r))(r \ll n)$，实际上还是 $\theta(n\log\ n)$
  - 没有重复关键码的情况，需要 n 个不同的编码来表示它们
  - 即 $d>=\log_r n$，在 $\Omega(\log\ n)$中

讨 论：

1. 本章讨论的排序算法都是基于数组实现的，可否应用于动态链表？性能上是否有差异？
2. 试总结并证明各种排序算法的稳定性，若算法稳定，如何修改可以使之不稳定？若算法不稳定，可否通过修改使之稳定？
3. 试调研STL中的各种排序函数是如何组合各种排序算法的。

数据结构与算法

谢谢聆听

国家精品课“数据结构与算法”
http://www.jpk.pku.edu.cn/pkujpk/course/sjjg/

张铭，王腾蛟，赵海燕
高等教育出版社，2008. 6。“十一五”国家级规划教材