

磁盘文件最优存储问题

2018年5月6日

1 题目介绍

1.1 问题描述

设磁盘上有 n 个文件 f_1, f_2, \dots, f_n , 每个文件占用磁盘上的1个磁道。这 n 个文件的检索概率分别是 p_1, p_2, \dots, p_n , 且 $\sum p_i = 1$ 。磁头从当前磁道移到被检信息磁道所需的时间可用这2个磁道之间的径向距离来度量。如果文件 f_i 存放在第 i 道上, $1 \leq i \leq n$, 则检索这 n 个文件的期望时间是对于所有的 i, j , $\text{time} += p_i * p_j * d(i, j)$ 。其中 $d(i, j)$ 是第 i 道与第 j 道之间的径向距离 $|i - j|$ 。

磁盘文件的最优存储问题要求确定这 n 个文件在磁盘上的存储位置, 使期望检索时间达到最小。

1.2 算法要求

对于给定的文件检索概率, 计算磁盘文件的最优存储方案。

1.3 数据输入输出

由文件input.txt给出输入数据。第一行为文件个数 n , 第二行为 n 个正整数表示文件的检索概率。将计算出的最小检索时间输出到文件output.txt。

2 题目分析

本题的贪心选择性质为, 使概率值大的两个文件尽可能的挨在一起。即先让文件按概率大小排序, 概率最大的应该放中间, 第二大和第三大的分别放在最大的左右两边, 第四大的放在第二大左边, 第五大的放在第三大的右边, 依次下去就行。

3 算法设计

将文件按概率排序后，用一个新的数组代表各文件存放位置，先确定mid值，再依次确定mid两边，mid两边的两边的各文件，不断的循环，最后各文件就再新的数组里放好了位置。

利用贪心算法实现如下算法：

```
double greedy(double p[], int n)
{
    qsort(p, n, sizeof(double), cmp);
    int mid = (n - 1) / 2; //数组从开始，所以是0n-1
    double pnew[1000];
    double sum = 0, exp = 0;
    pnew[mid] = p[n-1];
    for (int i = mid+1; i < n; i++)
        pnew[i] = p[n - 2 * (i - mid)];
    for (int i = mid - 1; i >= 0; i--)
        pnew[i] = p[n - 2 * (mid - i) - 1];
    for (int i = 0; i < n; i++)
    {
        sum += p[i];
        for (int j = i+1; j < n; j++)
            exp += pnew[i] * pnew[j] * (j - i);
    }
    return exp/sum/sum;
}
```

4 算法分析

4.1 复杂度

算法的复杂度分为求和和排序两个部分。求和复杂度为 $O(n^2)$ 。排序的复杂度为快速排序的复杂度，为 $O(n \log n)$ 。所以，最终算法的复杂度为 $O(n^2)$ 。