**4-4磁盘文件最优存储问题**

题意:设磁盘上有n个文件f1,f2,…,fn,每个文件占磁盘上1个磁道。这

n个文件的检索概率分别是p1,p2,…,pn,且p1+p2+…+pn  =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,fi存放在第i（1≤i≤n）道上 ，则检索这n 个文件的期望时间是 ∑ Pi\*Pj\*d（i，j）。式中 ，d（i，j）是第i道与第

1≤i≤j≤n

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。试设计一个解此问题的算法，并分析算法的正确性与计算复杂性。

对于给定的文件检索概率，编程计算磁盘文件的最优存储方案。

数据输入：

第1行是正整数 n，表示文件个数。第 2 行有 n 个正整数表a i表示文件的检

n

索概率。实际上第k个文件的检索概率应为a k /∑ai。

i=1

/\*\*@贪心算法-磁盘文件最优存储问题@ author-狂热的coder\*/

#include<iostream>

#include<algorithm>

using namespace std;

bool cmp(int a,int b)

{

return a>b;

}

/\*将数组a的值排序使其元素的分布从中间往两边依次减少\*/

void strageSort(int n,int a[]){

int i,k,mid;

sort(a,a+n,cmp);

mid=n/2;

int b[n+1];

b[mid]=a[0];

for(i=1,k=1;i<n;i++,k++){ //数组a的值分布从中间往两边依次减少

b[mid-k]=a[i];

i++;

if(i!=n)

b[mid+k]=a[i];

}

for(int i=0;i<n;i++){ //经变化后的a数组

a[i]=b[i];

}

}

void minStorage(int n,int a[]){

int sum=0;

for(int i=0;i<n;i++){

sum+=a[i];

}

double result=0;

for(int i=0;i<n;i++){

for(int j=i+1;j<n;j++){ //从磁道0-n-1。计算它们的磁道间的检索时间

result+=(a[i]\*1.0/sum)\*(a[j]\*1.0/sum)\*(j-i);

}

}

cout<<"最小期望检索时间:"<<result<<endl;

}

int main(){

int n,k,mid,i;

cout<<"输入文件个数"<<endl;

cin>>n;

int a[n+1];

cout<<"输入文件的检索概率:"<<endl;

for(int i=0;i<n;i++){

cin>>a[i];

}

strageSort(n,a);

minStorage(n,a);

return 0;

}

得到结果：

输入文件个数：

5

输入文件的检索概率：

9

11

22

33

55

最小期望检索时间:

0.547396