

# 磁带最优存储问题

## 算法思路

首先对于平均读取时间进行了如下的简化：

$$tr = c \sum_{k=1}^n p_{ik} l_{ik}$$
$$\sum_{r=1}^n t_r = C(n \cdot p_{i1} \cdot l_{i1} + (n-1) \cdot p_{i2} l_{i2} + \cdots + p_{in} \cdot l_{in})$$

可以看出越出现在前面的程序，调用的次数越多，由于想让平均的读取时间最小，所以考虑使用贪心策略，让越小的出现在越前面，即p\*l长度从小到大依次存放在磁带上。

## 算法实现

```
#include <iostream>

using namespace std;

typedef struct file_info{
    double pro;
    double len;
}file_info;

bool cmp(file_info file1, file_info file2){
    return file1.pro * file1.len < file2.pro * file2.len;
}

double min_cost(file_info *file, int num){
    double sum = 0;
    for (int i = 0; i < num; i++) {
        sum += (num - i) * file[i].pro * file[i].len;
    }
    return sum;
}

int main(int argc, const char * argv[]) {
    int num;
    cin >> num;
    int pro_sum = 0;
    file_info file[num];
    for (int i = 0; i < num; i++) {
        cin>> file[i].len >> file[i].pro;
        pro_sum += file[i].pro;
    }
    for (int i = 0; i < num; i++) {
```

```
        file[i].pro /= pro_sum;
    }
    sort(file, file + num, cmp);
    cout << "output: " << min_cost(file, num);
    return 0;
}
```

## 运行结果

---

```
5
71 872
46 452
9 265
73 120
35 85
output: 85.6193Program ended with exit code: 0
```