計算機程式及應用

H34996158 G6 侯瑋明

/\*

This code can be compiled and run ok.

Input the attachment to caculate how to bring goods to get the biggest profit.

1. Brute force

2. Greedy algorithm

usage (how to run):

./a (then input filename)

input file:

n4、n16、n32

output files:

none

compile (how to compile):

g++ -o fc hw4.cpp

pseudocode:

Input the given value and create arrays to record the caculation.

Through two methods "Brute force" and "Greedy algorithm", we can find the max profit combination.

Use binary law to judge the good to take or not to take so that we obtain the Brute force value.

Use bubble sort to sort the information so that we obtain the Greedy algorithm value.

coded by Wei-Ming Hou, group: 6, ID: h34996158, email: jacob0701@hotmail.com.tw

date: 2011.04.17

\*/

#include <iostream>

#include <math.h>

#include <ctime>

#include <fstream>

#include <cstdio>

using namespace std;

void initialize(const char \*, int \*, int \*, double \*\*, double \*\*, double \*\*, bool \*\*, bool \*\*, bool \*\*, int \*\*);

void Get\_Rorder(int \*\*, int, double \*\*);

int main()

{

int weight\_max, n, \*Rorder;

double \*w, \*p, \*Rpw;

bool \*x, \*XB, \*XG;

char filename[50];

cout << "Enter filename of graph: ";

cin >> filename;

initialize(filename, &n, &weight\_max, &w, &p, &Rpw, &x, &XB, &XG, &Rorder);

double weight\_sum, profit\_sum, profit\_max\_BF=0, profit\_max\_GA=0;

//-------------------------------------------------------------

time\_t t\_start, t\_end; //變數宣告

clock\_t c\_start, c\_end;

double t\_duration, c\_duration; //記錄時間長度

t\_start = time(NULL); //紀錄開始的時間

c\_start=clock();

//--暴力法-----------------------------------------------------

int c=1, d, i, k;

long long int b = static\_cast<long long int>(pow(2,n)-1); //計算2的n次

long long int a;

for(a=1; a<=b; a++)

{

c=1;

d=a;

weight\_sum=0;

profit\_sum=0;

while(d!=0) //2進位法

{

x[n-c]=d%2;

d=d/2;

c++;

}

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

{

weight\_sum += w[i] \* x[i];

profit\_sum += p[i] \* x[i];

}

if(weight\_sum>weight\_max) //若ture跳出迴圈

continue;

if(profit\_sum > profit\_max\_BF)

{

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

XB[i]=x[i];

profit\_max\_BF = profit\_sum;

}

}

//--貪婪法-----------------------------------------------------

weight\_sum=0;

int r;

for(i=0;i<n;i++) //計算利重比

{

Rpw[i]=p[i]/w[i];

}

Get\_Rorder(&Rorder, n, &Rpw);

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

{

if(weight\_sum>weight\_max)

break;

weight\_sum += w[Rorder[i]];

profit\_sum += p[Rorder[i]];

r=i;

}

for(i=0 ; i<r; i++)

profit\_max\_GA+=p[Rorder[i]];

//-------------------------------------------------------------

cout << "------ input file: " << filename << "------" << endl;

cout << "BF: Max Profit : " << profit\_max\_BF << endl;

cout << " Selected items : ";

for(int j=0; j<n; j++)

{

if(XB[j]==1)

cout << j << ' ';

}

cout << ' ' <<endl;

cout << "GA: Max Profit : " << profit\_max\_GA << endl;

cout << " Selected items : ";

for(i=0 ; i<r ; i++)

cout << Rorder[i] << ' ';

cout << ' ' <<endl;

//-------------------------------------------------------------

t\_end = time(NULL); //紀錄結束的時間

c\_end=clock();

t\_duration = difftime(t\_end, t\_start); //計算經過的時間，單位：秒

c\_duration = c\_end-c\_start;

cout << " duration by time\_t = " << t\_duration << " (s)" <<endl

<< " duration by clock\_t = " << c\_duration/CLK\_TCK << " (ms)" <<endl;

//--------------------------------------------------------------

system("pause");

return 0;

}

void Get\_Rorder(int \*\*Rorder,int n, double \*\*Rpw)

{

int i, j;

double tmp=0;

double \*text=new double[n];

for(i=0;i<n;i++) //令一迴圈以計算

{

text[i]=(\*Rpw)[i];

}

for(i=0;i<n;i++) //改變次序

{

for(j=i;j<n;j++)

{

if(text[i]<text[j])

{

tmp=text[i];

text[i]=text[j];

text[j]=tmp;

}

}

}

for(i=0 ; i<n; i++) //填入物品編號

{

for(j=0 ; j<n; j++)

{

if(text[i]==(\*Rpw)[j])

{

(\*Rorder)[i]=j;

(\*Rpw)[j]=0;

break;

}

}

}

}

void initialize(const char \*filename, int \*n, int \*weight\_max, double \*\*w, double \*\*p, double \*\*Rpw, bool \*\*x, bool \*\*XB, bool \*\*XG, int \*\*Rorder)

{

int i;

ifstream inFile(filename);

inFile >> (\*n);

inFile >> (\*weight\_max);

(\*w) = new double [(\*n)];

(\*p) = new double [(\*n)];

(\*Rpw) = new double [(\*n)];

(\*x) = new bool [(\*n)];

(\*XB) = new bool [(\*n)];

(\*XG) = new bool [(\*n)];

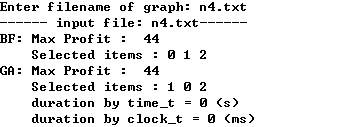
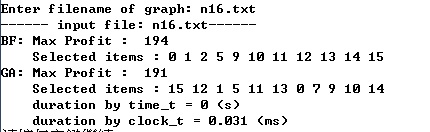
(\*Rorder) = new int [(\*n)];

for(i=0; i<(\*n); i++){

inFile >> (\*w)[i] >> (\*p)[i];}

inFile.close();

}

n4.txt n16.txt

n32.txt

