算法设计与分析第五次作业

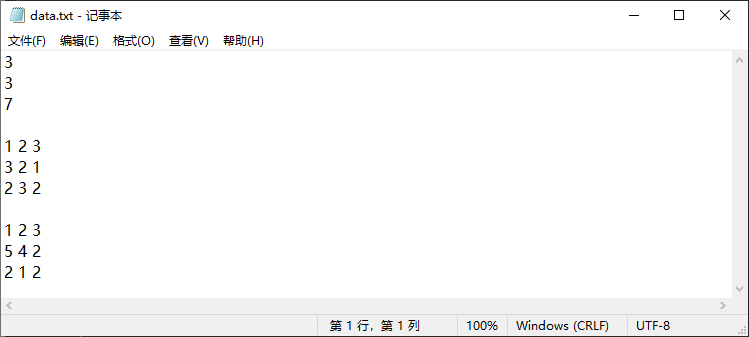
2019202243 郭炜

题目描述：

设某一机器由个部件组成，每一种部件都可以从个不同的供应商处购得。设 是从供应商处购得的部件的重量，是相应的价格。试设计一个算法，给出总价格不超过 的最小重量机器设计。

Code：

Data.txt



Source.cpp

代码可以在下面的github的url上查看，word格式不是很好

<https://github.com/SnailForce/The-Design-And-Analysis-Of-Algorithms/blob/master/6/Project1/Project1/Source.cpp>

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

#include <vector>

using namespace std;

constexpr auto MAX = 20;

unsigned int n, m;

vector<vector<int>> weight;

vector<vector<int>> value;

vector<int> bestX;

vector<int> x(MAX, 0);

unsigned int maxCost;

unsigned int curWeight, curValue, resWeight = 0xffffffff;

*// 递归*

void bfs(unsigned int *t*)

{

    if (t == n)

    {

        resWeight = curWeight;

        bestX.assign(x.begin(), x.end());

    }

    else

    {

        for (size\_t i = 0; i < m; i++)

        {

*// 判断成本是否超过预期以及当前重量是否超过最佳重量*

            if (curValue + value.at(t).at(i) <= maxCost && curWeight + weight.at(t).at(i) < resWeight)

            {

*// i从0开始，供货商从1开始*

                x.at(t) = i + 1;

                curWeight += weight.at(t).at(i);

                curValue += value.at(t).at(i);

                bfs(t + 1);

                curWeight -= weight.at(t).at(i);

                curValue -= value.at(t).at(i);

            }

        }

    }

}

int main()

{

*// 读取文件*

    string infile = "./data.txt";

    vector<string> data;

    ifstream is(infile, ifstream::in);

    if (is)

    {

        while (!is.eof())

        {

            string s;

            getline(is, s);

            if (s != "")

            {

                data.push\_back(s);

            }

        }

    }

    else

    {

        cout << "Unable to open file" << endl;

    }

    is.close();

    n = stoi(data.at(0));

    m = stoi(data.at(1));

    maxCost = stoi(data.at(2));

    cout << "n: " << n << " m: " << m << " cost: " << maxCost << endl;

*// 命令行输入*

*/\**

*cout << "n: ";*

*cin >> n;*

*cout << "m: ";*

*cin >> m;*

*cout << "cost: ";*

*cin >> maxCost;*

*\*/*

    for (size\_t i = 0; i < n; i++)

    {

        string cur = data.at(i + 3);

        stringstream input(cur);

        string res;

        vector<int> tmp;

        while (input >> res)

        {

            tmp.push\_back(stoi(res));

        }

        weight.push\_back(tmp);

    }

    for (size\_t i = 0; i < n; i++)

    {

        string cur = data.at(i + n + 3);

        stringstream input(cur);

        string res;

        vector<int> tmp;

        while (input >> res)

        {

            tmp.push\_back(stoi(res));

        }

        value.push\_back(tmp);

    }

    cout << "weight: " << endl;

    for (auto u : weight)

    {

        for (auto r : u)

        {

            cout << r << " ";

        }

        cout << endl;

    }

    cout << "value: " << endl;

    for (auto u : value)

    {

        for (auto r : u)

        {

            cout << r << " ";

        }

        cout << endl;

    }

    bfs(0);

    cout << "-- res --" << endl;

    cout << "best x: ";

    for (size\_t u = 0; u < n; u++)

    {

        cout << bestX.at(u) << " ";

    }

    cout << endl;

    cout << "best weight: " << resWeight << endl;

    return 0;

}

Result:

