**算法分析与设计实验报告**

**第 2 次实验**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 杨杰 | 学号 | 201908010705 | | 班级 | 计科1907 |
| 时间 | 4.23 | 地点 | 软件大楼 | | | |
| 实验名称 | 贪心算法解背包问题 | | | | | |
| 实验目的 | 通过上机实验，要求掌握贪心算法解背包问题的问题描述、算法设计思想、程序设计。 | | | | | |
| 实验原理 | 每次做出局部最优的选择，最后达成最优解。利用贪心算法解背包问题，根据不同数据规模的测试用例，能准确输出测试用例的最优解，并计算出程序运行所需要的时间。  选取的贪心选择策略为单位效益大的优先。 | | | | | |
| 实验步骤 | 1、依贪心选择策略，将尽可能多的价值最高的物品装入背包。 2、若将这种物品全部装入背包后，背包内的物品总重量未超过c，则选择价值次高的物品并尽可能多的装入背包。 3、依该策略一直进行下去，直到背包装满。 | | | | | |
| 关键代码 | void bbsort(vector<float> &w, vector<float> &v, int n)*//将物品按单位效益降序排列*  {      Thing T[n];      for (int i = 0; i < n; i++)      {          T[i].w = w[i];          T[i].v = v[i];          T[i].vperw = v[i] / w[i];      }      sort(T, T + n, cmp);      for (int i = 0; i < n; i++)      {          v[i] = T[i].v;          w[i] = T[i].w;      }  }  void beibao(vector<float> &w, vector<float> &v, int n, float c, float \*x)*//构造最优解*  {      bbsort(w, v, n);      int i;      for (i = 0; i < n; i++)      {          if (c < w[i])              break;          x[i] = 1;          c -= w[i];      }      if (i < n)          x[i] = c / w[i];  } | | | | | |
| 算法复杂度分析 | bbsort函数中for循环执行n次，时间复杂度为O(n),sort函数时间复杂度为O(nlogn),总时间复杂度为O(nlogn).  beibao函数中for循环最多执行n次，时间复杂度为O(n),bbsort函数时间复杂度为O(nlogn)，总时间复杂度为O(nlogn).  main函数中 for循环执行n次，时间复杂度为O(n)，while循环执行n次，时间复杂度为O(n)，beibao函数时间复杂度为O(nlogn)，所以最终的时间复杂度为O(nlogn). | | | | | |
| 测试结果  （含运行时间） | 小规模数据      中规模数据        大规模数据 | | | | | |
| 实验心得 | 贪心算法是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，算法得到的是在某种意义上的局部最优解。  贪心算法不是对所有问题都能得到整体最优解，关键是贪心策略的选择。  在实验中遇到了几点问题，现总结如下：     1. bbsort函数是用来将物品按单位效益降序排列的，但是发现按照上述写法并不能修改vector的值，这与数组传参是不同的，要想修改vector的值需要传引用。修改如下：      1. 背包问题与0-1背包问题的区别在于x[i]的取值，0-1背包x[i]只能取0或1，而背包问题x[i]可以取0到1中的任何数，也就是说可以只装物品的一部分。这也导致0-1背包问题不能用贪心算法，而应该用动态规划。所以在背包问题中x[]数组的类型也要进行修改，不再是int,而是float。     经过这次实验，我对于贪心算法解背包问题的相关代码已基本熟悉，算法知识得到了复习与巩固。在写代码与调试的过程中，在解决问题过程中，丰富了个人编程的经历和经验，提高了个人解决问题的能力。 | | | | | |
| 实验得分 |  | 助教签名 | |  | | |

**附录：完整代码**

*//贪心法实现背包问题*

#include <iostream>

#include <fstream>

#include <windows.h>

#include <algorithm>

#include <vector>

using namespace std;

struct Thing

{

    float w;

    float v;

    float vperw;

};

bool cmp(Thing l, Thing r)

{

    return l.vperw > r.vperw;

}

void bbsort(vector<float> &w, vector<float> &v, int n) *//将物品按单位效益降序排列*

{

    Thing T[n];

    for (int i = 0; i < n; i++)

    {

        T[i].w = w[i];

        T[i].v = v[i];

        T[i].vperw = v[i] / w[i];

    }

    sort(T, T + n, cmp);

    for (int i = 0; i < n; i++)

    {

        v[i] = T[i].v;

        w[i] = T[i].w;

    }

}

void beibao(vector<float> &w, vector<float> &v, int n, float c, float \*x)*//构造最优解*

{

    bbsort(w, v, n);

    int i;

    for (i = 0; i < n; i++)

    {

        if (c < w[i])

            break;

        x[i] = 1;

        c -= w[i];

    }

    if (i < n)

        x[i] = c / w[i];

}

int main()

{

    float c = 20;*//背包容量*

    float a, b;

    vector<float> w, v;

    ifstream infile("beibao\_in\_s.txt", ios::in);

    while (infile >> a >> b)

    {

        w.push\_back(a);

        v.push\_back(b);

    }

    int n = v.size();

    float x[n];

    memset(x, 0, sizeof(float) \* n);*//x[]数组初始化为0*

    ofstream outfile("beibao\_out\_s.txt", ios::out);

    double time = 0;

    LARGE\_INTEGER nFreq, nBeginTime, nEndTime;

    QueryPerformanceFrequency(&nFreq);

    QueryPerformanceCounter(&nBeginTime);*//开始计时*

    beibao(w, v, n, c, x);

    float sum = 0;

    for (int i = 0; i < n; i++)*//计算最优值*

    {

        sum += x[i] \* v[i];

    }

    outfile << sum << endl;

    for (int i = 0; i < n; i++)

    {

        outfile << x[i] << ' ';

        if (i != 0 && i % 20 == 0)

            outfile << endl;

}

outfile << endl;

    QueryPerformanceCounter(&nEndTime);*//停止计时*

    time = (double)(nEndTime.QuadPart - nBeginTime.QuadPart) / (double)nFreq.QuadPart;*//计算程序执行时间单位为s*

    outfile << "程序耗时" << time \* 1000 << "ms" << endl;

    infile.close();

    outfile.close();

    return 0;

}

*//背包数据生成器*

#include <iostream>

#include <fstream>

#include <time.h>

#define random(a, b) (rand() % (b - a + 1) + a)

using namespace std;

int main()

{

    int t;

    cin >> t;

    ofstream outfile("beibao\_in\_m.txt", ios::out);

    srand((int)time(NULL));

    for (int i = 1; i <= 2 \* t; i++)

    {

        outfile << random(1, 100) << " ";

        if (i % 20 == 0)

            outfile << endl;

    }

    outfile.close();

    return 0;

}