

排序贪心

T E A C H I N G C O U R S E W A R E P O W P O I N T

授课时间：2025.07.30

目录

● PART-01 算法引入 TEACHING
COURSEWARE

● PART-02 概念分析 TEACHING
COURSEWARE

● PART-03 经典例题 TEACHING
COURSEWARE

● PART-04 Q&A TEACHING
COURSEWARE

01

算法引入

TEACHING
COURSEWARE

TEACH



算法引入

2 5 1 3 9

02

题型分析

TEACHING
COURSEWARE

TEACH



通用模版

找到一个合理的排序规则，然后按顺序做局部最优选择。

需要通过排序给所有元素定一个“优先级”，再按照优先级依次选择，最终得到全局最优。

03

经典例题

TEACHING
COURSEWARE

TEACH

题目描述

[复制 Markdown](#) [展开](#) [进入 IDE 模式](#)

阿里巴巴走进了装满宝藏的藏宝洞。藏宝洞里面有 N ($N \leq 100$) 堆金币，第 i 堆金币的总重量和总价值分别是 m_i, v_i ($1 \leq m_i, v_i \leq 100$)。阿里巴巴有一个承重量为 T ($T \leq 1000$) 的背包，但并不一定有办法将全部的金币都装进去。他想装走尽可能多价值的金币。所有金币都可以随意分割，分割完的金币重量价值比（也就是单位价格）不变。请问阿里巴巴最多可以拿走多少价值的金币？

输入格式

第一行两个整数 N, T 。

接下来 N 行，每行两个整数 m_i, v_i 。

输出格式

一个实数表示答案，输出两位小数

输入输出样例

输入 #1

[复制](#)

```
4 50
10 60
20 100
30 120
15 45
```

输出 #1

[复制](#)

```
240.00
```




经典例题

贪心。

所有金币都可以分开，也就是说只要按照**性价比**最高的取一定得到的价值最大。

性价比就是这堆金币的**价值除以重量**。

只需要把这n堆金币按性价比排序就行了。

然后依次遍历，如果背包中剩余可以拿的重量大于等于这堆金币的重量，就全拿，否则直接装满。

经典例题

```
#include<bits/stdc++.h>
using namespace std;
struct Node {
    int w, v; // w 表示重量, v 表示价值
} a[110];
// 按性价比 (价值/重量) 从高到低排序
bool cmp(Node aa, Node bb) {

    return (double)aa.v / aa.w > (double)bb.v / bb.w;
}
```

部分背包

```
// 读取每种金币的重量和价值
for (int i = 1; i <= n; i++) {
    cin >> a[i].w >> a[i].v;
}

// 按性价比排序 (从高到低)
sort(a + 1, a + n + 1, cmp);

// 贪心策略: 优先选择性价比高的金币
for (int i = 1; i <= n; i++) {
    if (a[i].w <= m) {
        // 当前金币可以全部装入背包
        ans += a[i].v;
        m -= a[i].w;
    } else {
        // 当前金币只能部分装入背包
        ans += a[i].v * m * 1.0 / a[i].w;
        break; // 背包已满, 直接退出循环
    }
}

// 输出结果, 保留两位小数
printf("%.2lf", ans);
return 0;
```

04

Q&A

TEACHING
COURSEWARE

TEACH



Q & A

Q: 我们为什么要排序