

贪心基础

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



排序 (sort)

一、什么是 sort 函数？

sort, 排序, 告诉 sort 函数 “从哪开始” “到哪结束” “按什么顺序排”, 它就会帮你搞定。

举个例子: 你有 4 个数字: 3、1、2、4, 想从小到大排好。用 sort 函数后, 它会自动变成 1、2、3、4、

`sort(a+1, a+5+1)`

二、怎么用 sort 函数？

第一步：准备好要排序的 “东西”

比如: `int a[5] = {3, 1, 5, 2, 4};` 这个a就是装数字的盒子, 里面有 5 个数字。

第二步：告诉 sort 函数范围

sort 函数需要知道 “从盒子的第几个开始” 和 “到第几个结束”。

在代码里, 开头用a+数字, 结束用a+数字。

比如盒子里有 5 个数字, 就写

`sort(a, a+5)`



排序 (sort)

默认情况下，sort 会按 “从小到大” 排。如果想从大到小排，就要加一个 “比较器” —— 就像告诉它 “按从大到小的规则来”。

这个 “比较器” 是一个简单的函数，比如：

```
bool cmp(int x, int y) {  
    return x > y; // 意思是 “x比y大的话，x排在前面”  
}
```

然后把这个函数名加到 sort 里：sort(a, a+5, cmp)，就能从大到小排了。

```
sort(a, a+5, cmp);
```

```
struct Student {  
    string name;  
    int height;  
};
```

// 比较器：按身高从矮到高

```
bool cmp(Student s1, Student s2)  
{  
    return s1.height < s2.height;  
}
```

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

钱

02

概念分析

TEACHING
COURSEWARE

TEACH

贪心算法是一种在**每一步**决策时，都选择**当前状态下最优**（即最有利）的选项，且**不回溯**修改之前决策的算法。

它的核心思路可以概括为：“走一步看一步，每一步都选当下最好的，**不考虑未来**可能的变化，也**不后悔之前**的选择”。

重要性质：“**无后效性**”

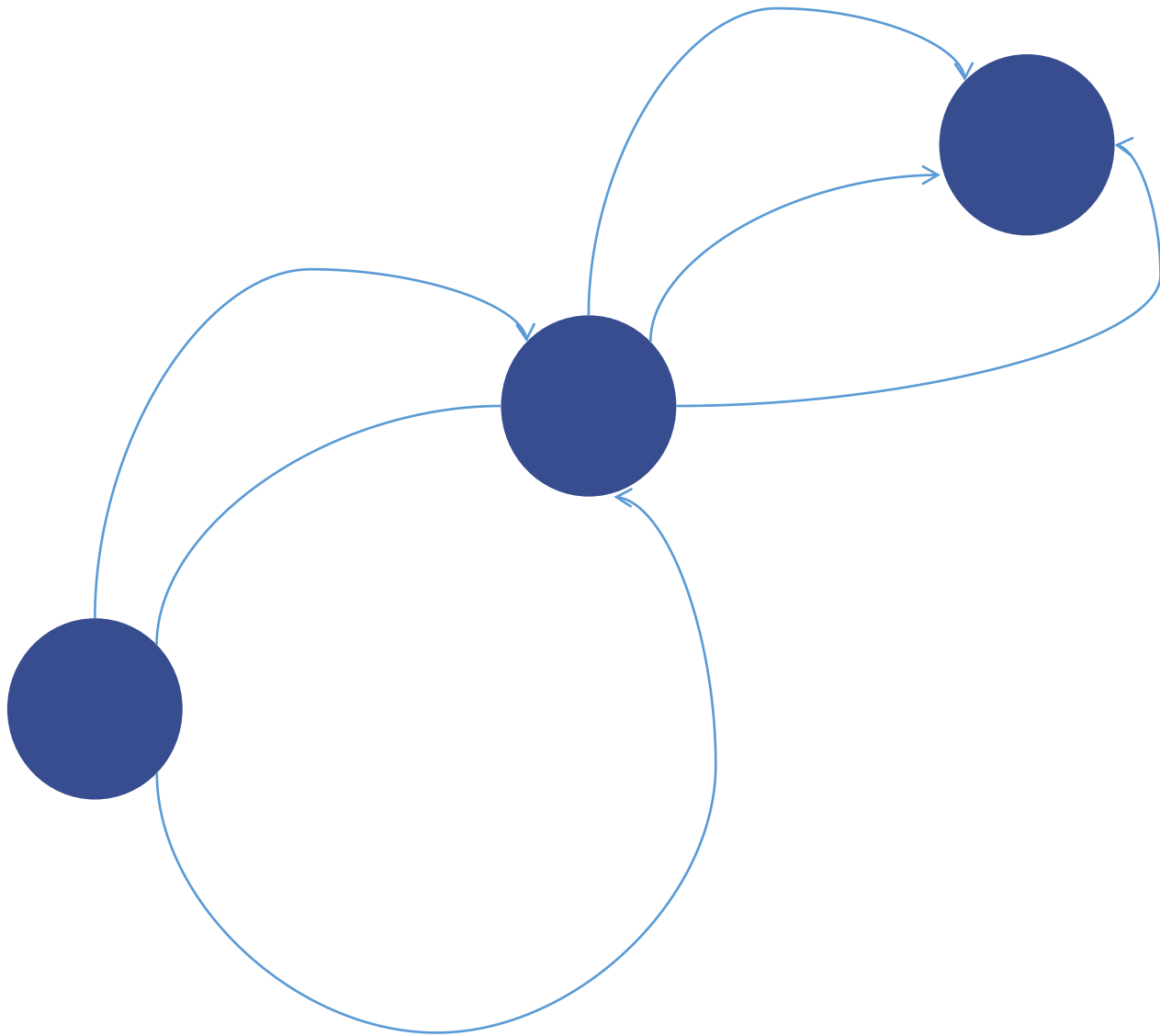
关键性质：

贪心选择性质：局部最优能推出全局最优。

局部和全局

重点！

概念分析





概念分析

每一步的选择仅由当前的局部最优状态决定，且一旦做出选择，就不会被后续的任何决策所改变或否定。后续的所有决策都基于当前选择后的状态继续进行，无需回溯调整之前的选择。

03

经典例题

TEACHING
COURSEWARE

TEACH

经典例题

题目描述

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

有 n 个人在一个水龙头前排队接水，假如每个人接水的时间为 T_i ，请编程找出这 n 个人排队的一种顺序，使得 n 个人的平均等待时间最小。

如果两个人接水的时间相同，编号更小的人应当排在前面。

输入格式

第一行为一个整数 n 。

第二行 n 个整数，第 i 个整数 T_i 表示第 i 个人的接水时间 T_i 。

输出格式

输出文件有两行，第一行为一种平均时间最短的排队顺序；第二行为这种排列方案下的平均等待时间（输出结果精确到小数点后两位）。

说明/提示

$1 \leq n \leq 1000$, $1 \leq t_i \leq 10^6$ ，不保证 t_i 不重复。



经典例题

样例输入：

4

1 2 3 4

样例输出：

2.50

经典例题

贪心策略：

核心思想：让接水时间短的人先接水。

为什么？

直觉上就能感受到：让耗时少的人快点结束，后面的人就不用等那么久。
一个耗时很长的人在前面，会让后面所有人等很久。

为什么？

这个局部最优策略（每次选当前剩余人中接水时间最短的）直接导致了全局最优解（**总等待时间**最小）。

经典例题

```
#include <bits/stdc++.h>
using namespace std;
struct node
{
    int b, num; // b存储接水时间, num存储原始序号
}a[1010];
bool cmp(node x, node y) // 排序规则
{
    return x.b < y.b;
}
```

```
int main()
{
    int n, i, j;
    double time = 0;

    cin >> n;
    for(i = 1; i <= n; i++)
    {
        cin >> a[i].b;
        a[i].num = i; // 存储原始序号
    }

    sort(a + 1, a + n + 1, cmp); // 按接水时间排序

    for(i = 1; i <= n; i++)
    {
        cout << a[i].num << " "; // 输出排序后的原始序号
    }
    cout << endl; // 换行

    for(j = n - 1; j >= 1; j--) // 计算总等待时间, j为等待人数
    {
        i = n - j; // 当前接水的人在排序后的位置
        time += a[i].b * j; // 累加等待时间, 当前人的接水时间乘以等待人数
    }

    printf("%.2lf", time / n); // 输出平均等待时间, 保留两位小数

    return 0;
}
```

04

Q&A

TEACHING
COURSEWARE

TEACH



Q & A

Q：贪心最重要的性质：

Q：贪心的关键性质：

Q：说说你对贪心的理解：