

算法初步

Ben

Why ?

- 应用：机器学习、数据挖掘、自然语言处理、密码学、计算机图形学等
- 研究：时空复杂度等
- 找工作：贪心、分治、动态规划、树、图等

What ?

- 大象放进冰箱
- 有穷性
 - 必须在人类毁灭前结束
- 确定性
 - 老板，便宜一点（ 康盲，一点是多少 ）
- 可行性
 - 造个飞碟（ 你这么厉害你家里人造么 ）
- 输入&输出

How ?

- 穷举（万能算法）

求N个数的全排列

8皇后问题

- 分而治之（减而治之）

二分查找——减而治之

归并排序——分而治之

How ?

- 贪心

最小生成树 Prim, Kruskal

单源最短路 Dijkstra

- 动态规划

背包

士兵路径

复杂度

- 谈算法不谈复杂度 = 耍流氓
- 硬件发展，速度提升、内存提升（常数级）
- 在实现之前，我们要预估算法所需要的资源
 - 时间
 - 空间

复杂度

- 时空复杂度

使用大O记号

时间：基本操作次数（汇编指令条数）

空间：占用内存字节数

区别：空间可以再利用

时空互换（Hash表）

复杂度

- 常见时间复杂度分析方法

数循环次数

均摊分析

递归式——主定理

复杂度

- $O(1)$

基本运算, $+$, $-$, $*$, $/$, $\%$, 寻址

- $O(\log n)$

二分查找

- $O(n^{1/2})$

枚举约数

复杂度

- $O(n)$

线性查找

- $O(n^2)$

朴素最近点对

- $O(n^3)$

Floyd最短路

普通矩阵乘法

复杂度

- $O(n \log n)$

归并排序

快速排序的期望复杂度

基于比较排序的算法下界

- $O(2^n)$

枚举全部的子集

复杂度

- $O(n!)$

枚举全排列

- 总结：

优秀 $O(1) < O(\log n) < O(n^{1/2}) < O(n) < O(n \log n)$

可能可以优化 $O(n^2) < O(n^3) < O(2^n) < O(n!)$

均摊分析

- 多个操作，一起算时间复杂度

MULTIPOP的队列，可以一次性出队 k 个元素

每个元素只出入队列一次

动态数组尾部插入操作（vector）

一旦元素超过容量限制，则扩大一倍，再复制

例题：最大子数组和

- 给定数组 $a[1\dots n]$ ，求最大子数组和，即找出 $1 \leq i \leq j \leq n$ ，使 $a[i] + a[i+1] + \dots + a[j]$ 最大
- 介绍三个算法
 - 暴力枚举 $O(n^3)$
 - 优化枚举 $O(n^2)$
 - 贪心法 $O(n)$

暴力枚举

- 暴力枚举：三重循环

for $i \leftarrow 1$ to n

 for $j \leftarrow i$ to n

$\text{sum} \leftarrow a[i] + \dots + a[j]$

$\text{ans} \leftarrow \max(\text{ans}, \text{sum})$

时间复杂度 $O(n^3)$ 附加空间复杂度 $O(1)$

优化枚举

- 优化枚举：两重循环

for $i \leftarrow 1$ to n

$sum \leftarrow 0$

 for $j \leftarrow i$ to n

$sum \leftarrow sum + a[i]$

$ans \leftarrow \max(ans, sum)$

时间复杂度 $O(n^2)$ 附加空间复杂度 $O(1)$

贪心法

- 贪心法：一重循环

$\text{sum} \leftarrow 0 \quad \text{ans} \leftarrow 0$

for $i \leftarrow 1$ to n

$\text{sum} \leftarrow \text{sum} + a[i]$

$\text{ans} \leftarrow \max(\text{sum}, \text{ans})$

if $(\text{sum} < 0)$ $\text{sum} \leftarrow 0$

时间复杂度 $O(n)$ 附加空间复杂度 $O(1)$

作业题1

- 设计一个队列
- 支持：出队，入队，求最大元素
- 要求 $O(1)$
- 均摊分析

作业题2

- 给定一个正整数数组 a ，是否能以3个数为边长构成三角形？
- 即是否存在不同的 i, j, k ,
- 满足 $a[i] < a[j] + a[k]$
- 并且 $a[j] < a[i] + a[k]$
- 并且 $a[k] < a[i] + a[j]$