



平衡规划

—— 浅析一类平衡思想的应用
福建省福州第八中学 郑 瞰

引言

平衡 → 状态 稳定

构建平衡



引言

➤ 经典算法的非典型实现 ✓

➤ 例题优秀的解法配美算法

➤ 修改主算法
平衡二叉树的简单构造

平衡二叉树的构造
平衡二叉树的构造
平衡二叉树的构造

平衡二叉树的构造

例题五：树的维护

特例与一般的转化

平衡特例与一般的关系

经典算法的非典型实现

➤经典算法：
有效解决一类问题

经典 \neq 简单

障碍

平衡规划

经典算法的非典型实现

实现不同→性能不同

➤ Jackpot (PKU 2103)

➤ 题目描述:

➤ 等概率选择任意整数，求选到的数能被给定的 p_1, p_2, \dots, p_n 中至少一个数整除的概率（用最简分数表示， $n \leq 16$ ）。

$$P = \lim_{k \rightarrow \infty} \left(\frac{S_k}{2k + 1} \right)$$

经典算法的非典型实现

实现不同 → 性能不同

2014年百度高精度实现
常见优化：压位
有效减少数组长度

```
for i ← 1 to a[0] + 1 do  
begin  
    p ← a[i] * b + p shv 10^k  
    a[i] ← p mod (2^k; 1);  
end;
```



输出速度
计算速度



经典算法的非典型实现

➤ 经典算法

有效解决一类问题。

➤ 经典 \neq 简单

难思 \rightarrow 简单
难实 \rightarrow 实用

经典算法的非典型实现

- 警卫安排问题 (ural 1099)
- 题目描述:
- 给定若干警卫间搭档关系，要求成对给警卫安排保卫工作，求能够安排警卫的最大值。（警卫人数不超过222）
联想：二分图最大匹配

经典算法：匈牙利树匹配！

经典算法：带花匈牙利树

经典算法的非典型实现

➤ 反例



经典算法的非典型实现

主要问题：无法找到可调整的
交错增广树。

解决方法：随机搜索序；
多次运行；

问题解决？
进一步优化：设置失败上限
提高单次运行最优解出现概率



经典算法的非典型实现

	初始算法			进一步优化后的算法		
运行次数\ 失败上限	10-1	20-1	50-1	5-5	5-10	20-10
Accepted	0%	60%	100%	90%	100%	0%
Wrong answer	100%	40%	0%	10%	0%	0%
Time limit exceeded	0%	0%	0%	0%	0%	100%
平均 AC 时间	----	0.039	0.080	0.076	0.147	----

数据来源: Ural1099

时限: 0.5 秒

经典算法的非典型实现

任意图匹配

难在保持时间复杂度低

思想本质：
削弱当前优势，平衡多部分间关系

总结

构建平衡

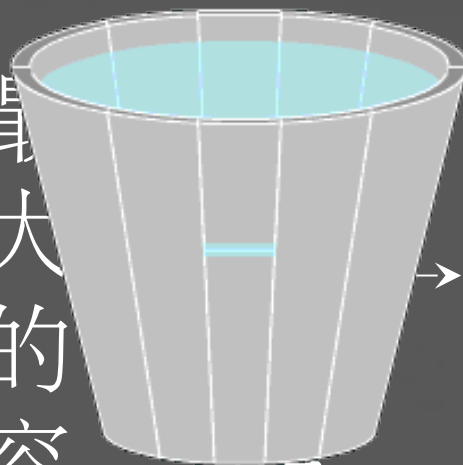
木桶理论

弱点 vs. 优势

最短的木板

决定

最大的容量



容量最大

总结

经典 \neq 简单

平衡 \neq 万能

- 实践才是感悟这一类问题最有效的途径。
- 纸上得来终觉浅，绝知此事要躬行。

谢谢

效果优秀的非完美算法

➤ NOI2007 Catch

➤ 题目描述:

➤ 给定树状公路土图，要求制定一个抓捕大盗 Frank 的方案。你的操作有：

- 1、在某个城市空降一位警探。
- 2、把留在某个城市里的一位警探直接召回指挥部。
- 3、让待在某个城市的一位警探沿着公路移动到另一个城市。

效果优秀的非完美算法

- 类似树状 DP 的贪心
- 存在反例。
- 实际的解给部分的解提供了一个弹性空间。
- 1000 组随机数据：
 - 约 85% : 5 个警探,
 - 约 15% : 6 个警探
- 随机数据: 至少 60% 的分数

复杂问题的简单化构造

- 树的维护 (PKU 3237)
- 题目描述:
- 给定一棵有 N 个节点的树，点编号为 $1 \sim N$ ，边按给定顺序编号为 $1 \sim N-1$ ，每一条边有一个边权。需要对这棵树维护三个操作：
 - 1、将某一条边的权值修改为 V 。
 - 2、将点 A 到 B 的路径上的边权值取负号。
 - 3、询问点 A 到 B 的路径上的权值最大值。数据组数不超过 20，点个数不超过 10000。

复杂问题的简单化构造

- 主要方法：拆链 + 线段树维护
- 拆链准则：自下而上，链长度尽量长。
- 小优化：边权下放至点。
- 最多拆链 $O(\sqrt{n})$
- 最坏时间复杂度 $O(n \log n + q \sqrt{n} \log n)$

复杂问题的简单化构造

- [证明]
- 集合 S ，它的元素是 x 到 $root$ 之间所有的边。
- 设 S 中的边属于 k 条拆分链，这些拆分链与路径公共部分的最上层节点为 $P[1]$ ，..... $P[k]$ ，其中 $P[k]=root$ 。
- $P[2]$ 必然连接着另外一条向下走的拆分链，这条拆分链除去属于 S 的部分至少还有 1 个点； $P[3]$ 必然连接着另外一条向下走的拆分链，这条拆分链除去属于 S 的部分至少还有 2 个点；.....； $P[k]$ 连接的向下走的拆分链除去属于 S 的部分至少还有 $k-1$ 个点。
- 最少点数： $1+2+3+...+k-1+k = \frac{k(k+1)}{2}$
- 最多点数： n

$$k = \frac{\sqrt{8n+1}-1}{2} = O(\sqrt{n})$$

经典算法的非典型实现

- 随机搜索序
- 表面：根据克制数据的顺序依赖性予以回避。
- 实质：设置随机权值，改变状态属性，回避特殊情况的生成。
- 对于本题：回避克制数据的生成。
- 适用范围：
- 特殊情况依赖一定权值属性的情况。
- 其他实例：Treap，随机快排

经典算法的非典型实现

- 设置失败上限
- 随机算法还是有可能无法得到我们理想的状态或结果。
- 提高准确率的一个优化。
- 通过对于局部问题的精益求精来强化全局最优解出现概率。
- 实例：大素数的测试。