

钻石矿工算法实验报告

- 班级-2013211313
- 姓名-彭俊喆，李雄伟，兰玉鹂
- 学号 2013211535,2013211536,2013211540
- 分工情况-代码（兰玉鹂） 文档（李雄伟） 测试（彭俊喆）
- 完成日期: 2015-5-20

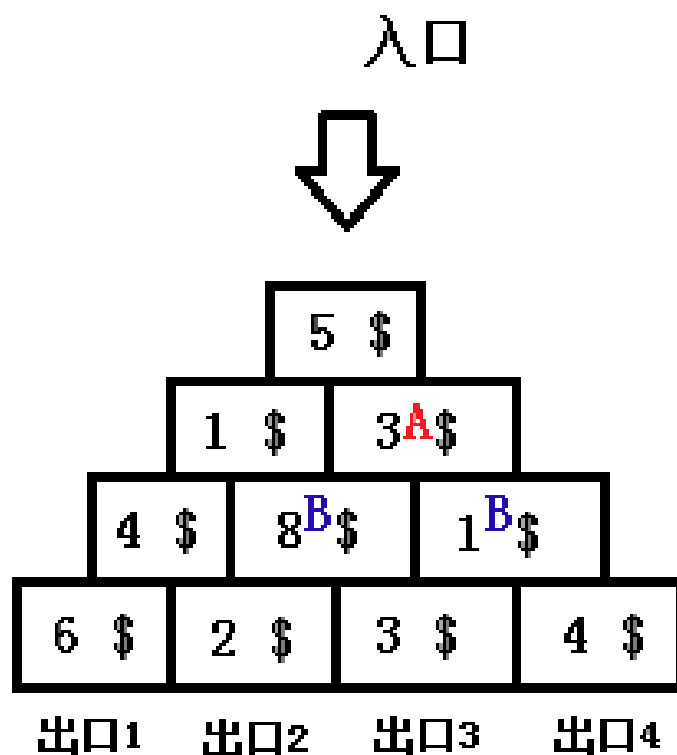
问题描述-

有一座金字塔

每块石头上都镶有一块钻石

从金字塔的顶端向下收集钻石尽可能收集价值高的钻石

只能从一块砖斜向左下或斜向右下走到另一块砖上



自我描述：对于钻石矿的挖取， 可以利用递归的观点来看， 在矿顶的位置， 无需选取， 但其往下的层数， 每次都需要选择最大的矿数即选择以下层矿为矿顶的次级金字塔， 如此递归可以很清晰的看到算法的本质就是要将此递归算法进行优化， 尽量压缩时间复杂度。

最优子结构：

采用一个数组 `int gold[1000][1000]` 来存储输入的每层中每个矿的矿数；另外用 `int maxGold[1000][1000]` 来存储每层中每个矿的子金字塔的最大矿数；`bool route[1000][1000]` 来存储每层中被选中的目标，即路线。

算法复杂度：

时间复杂度： $O(n^2)$ ；空间复杂度： $O(n^2)$ ；

对于分治法的时间复杂度： $O(2^n)$ ；

很明显 该算法的优化程度 要超过 分治法；

算法思想：

利用迭代来替换递归，为了获得每层钻石金字塔的最大矿石量的目标，必须知道该层以下的子金字塔的最大矿石量，即本层每个目标的最大矿石量以该目标为矿顶的子金字塔的最大矿石量 $i + \text{Max}(i-1)$ ，采用自底向上的策略。

设计描述

(1)自定义 `int getMaxGold(); void printRoute()`

(2)主函数 `int main();`

(3)算法描述（类语言）；

源代码：

`miner.cpp`

测试结果

```
请输入金字塔层数<最少为1层>:3
请输入每个砖块中的钻石数目:3 4 2 7 5 3
最大钻石数量为: 14
路径为: 3->4->7

-----
Process exited with return value 0
Press any key to continue . . .
```

```
请输入金字塔层数<最少为1层>:4
请输入每个砖块中的钻石数目:5 4 3 6 2 7 8 1 0 3
最大钻石数量为: 23
路径为: 5->4->6->8

-----
Process exited with return value 0
Press any key to continue . . .
```

程序运行无误

用户使用说明

程序运行环境: window 系统

程序启动方法: 直接运行 exe;

操作体会: 学习动态规划, 并学会对算法进行优化

心得体会

算法思路主要来自算法课上的同学的讲解, 同学的讲解深入浅出, 很容易理解他们的思路, 让我们对矿工算法的本质愈加清晰。

