



链滴

ARTS 012

作者: [lucianolixin](#)

原文链接: <https://ld246.com/article/1574063081880>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



ARTS 是由左耳朵耗子陈皓在极客时间专栏《左耳听风》中发起的一个每周学习打卡计划。

Algorithm: 至少做一个 LeetCode 的算法题。主要为了编程训练和学习。

Review : 阅读并点评至少一篇英文技术文章。主要为了学习英文, 如果你英文不行, 很难成为技术手。

Tip: 学习至少一个技术技巧。主要是为了总结和归纳你日常工作中所遇到的知识点。

Share: 分享一篇有观点和思考的技术文章。主要为了输出你的影响力, 能够输出你的价值观。

Algorithm

给定一个三角形, 找出自顶向下的最小路径和。每一步只能移动到下一行中相邻的结点上。

例如, 给定三角形:

```
[
  [2],
 [3,4],
 [6,5,7],
 [4,1,8,3]
]
```

自顶向下的最小路径和为 11 (即, $2 + 3 + 5 + 1 = 11$) 。

来源: 力扣 (LeetCode)

链接: <https://leetcode-cn.com/problems/triangle>

//自顶向下记忆化搜索

```
func minimumTotal(triangle [][]int) int {
    storeMem := make(map[string]int,)
    return findMax(triangle,0,0,storeMem)
}

func findMax(triangle [][]int,level int,index int,storeMem map[string]int) int{
    key :=strconv.Itoa(level)+"-"+strconv.Itoa(index)
    if value,ok := storeMem[key]; ok {
        return value
    }
    if level == len(triangle)-1 {
        storeMem[key] = triangle[level][index]
        return triangle[level][index]
    }

    left := findMax(triangle,level+1,index,storeMem)
    right := findMax(triangle,level+1,index+1,storeMem)
    if left < right {
        storeMem[key] = left+triangle[level][index]
        return left+triangle[level][index]
    }else{
        storeMem[key] = right+triangle[level][index]
        return right+triangle[level][index]
    }
}

//DP方程
// dp[i][j] = min(dp[i+1][j],dp[i+1][j+1])+dp[i][j]
```

```
func minimumTotal(triangle [][]int) int {
    //初始化状态数组
    dp := triangle
    for i:=len(triangle)-2 ; i>=0 ;i-- {
        for j:= 0 ; j < len(triangle[i]); j++ {
            if dp[i+1][j] < dp[i+1][j+1] {
                dp[i][j] += dp[i+1][j]
            }else{
                dp[i][j] += dp[i+1][j+1]
            }
        }
    }
    return dp[0][0]
}
```

Tip

面试技巧: 审题能力, 挖掘已知条件, 确定边界

需求不明确，范围不清晰的问题，很难得到合理的设计。

比如设计秒杀，那么预估它的qps是多少，不同量级的产品，设计出来的架构复杂程度会有很大不同。

另外就是为了考察候选人的产品分析能力，需要在提问的问题的基础之上进一步明确需求，确定边界（产品往往不会把所有需求说清楚）。