在一条直线上有n堆石子，每堆有一定的数量，每次可以将两堆相邻的石子合并，合并后放在两堆的中间位置，合并的费用为两堆石子的总数。求把所有石子合并成一堆的最小花费(定义dp[i][j]为第i堆石子到第j堆合并的最小花费)。

(1)写出该问题的递推方程。（10分）

(2)有5堆石子(n=5)，每堆石子大小分别为<1,3,5,2,4>,求出把所有石子合并成一堆的最小花费(要求写出运算矩阵)。（10分）

(3)写出该问题的伪代码。（10分）

答案：

(1)递推方程：

(2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** |
| **1** | 0 | 4 | 13 | 22 | 34 |
| **2** |  | 0 | 8 | 17 | 28 |
| **3** |  |  | 0 | 7 | 17 |
| **4** |  |  |  | 0 | 6 |
| **5** |  |  |  |  | 0 |

最小的合并花费为34

(3)伪代码：**p为石子数量的数组**

**Stone-merge (*p*)**

***n*=length(*p*)；**

**FOR *i=1* TO *n* DO**

***dp[i, i]=0*;**

**FOR *l=2* TO *n* DO**

**FOR *i=1* TO *n-l+1* DO**

***j=i+l-1*;**

***dp[i, j]=∞;***

**FOR *k←i* To *j-1* DO**

***q=dp[i, k]+dp[k+1, j]+sum[i,j]***

**IF *q<dp[i, j]*  THEN *dp[i,j]=q*;**

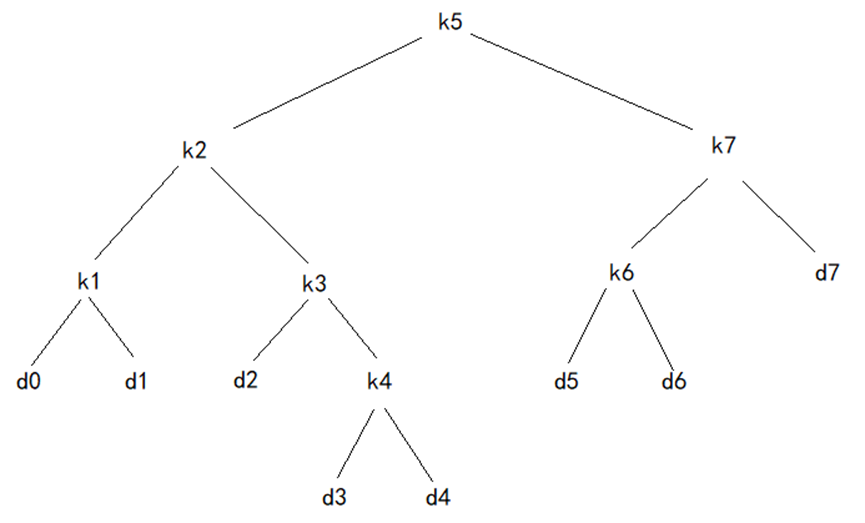
**Return dp;**

1. 若7个关键字的概率如下所示，求其最优二叉搜索树的结构和代价，要求写出递推方程。（30分）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  | 0.04 | 0.06 | 0.08 | 0.02 | 0.10 | 0.12 | 0.14 |
|  | 0.06 | 0.06 | 0.06 | 0.06 | 0.05 | 0.05 | 0.05 | 0.05 |

答案：代价为3.12

k5是根节点，k2是k5的左孩子，k1是k2的左孩子，d0是k1的左孩子，d1是k1的右孩子，k3是k2的右孩子，d2是k3的左孩子，k4是k3的右孩子，d3是k4的左孩子，d4是k4的右孩子，k7是k5的右孩子，k6是k7的左孩子，d5是k6的左孩子，d6是k6的右孩子，d7是k7的右孩子。



1. 编程题：兑换零钱问题（40分）

题目描述：

给定不同面额的硬币 coins 和一个总金额 amount。编写一个函数来计算可以凑成总金额所需的最少的硬币个数。如果没有任何一种硬币组合能组成总金额，返回-1。（提示：你可以认为每种硬币的数量是无限的）。

示例 1:

输入: coins = [1, 2, 5], amount = 11

输出: 3

解释: 11 = 5 + 5 + 1

示例 2:

输入: coins = [2], amount = 3

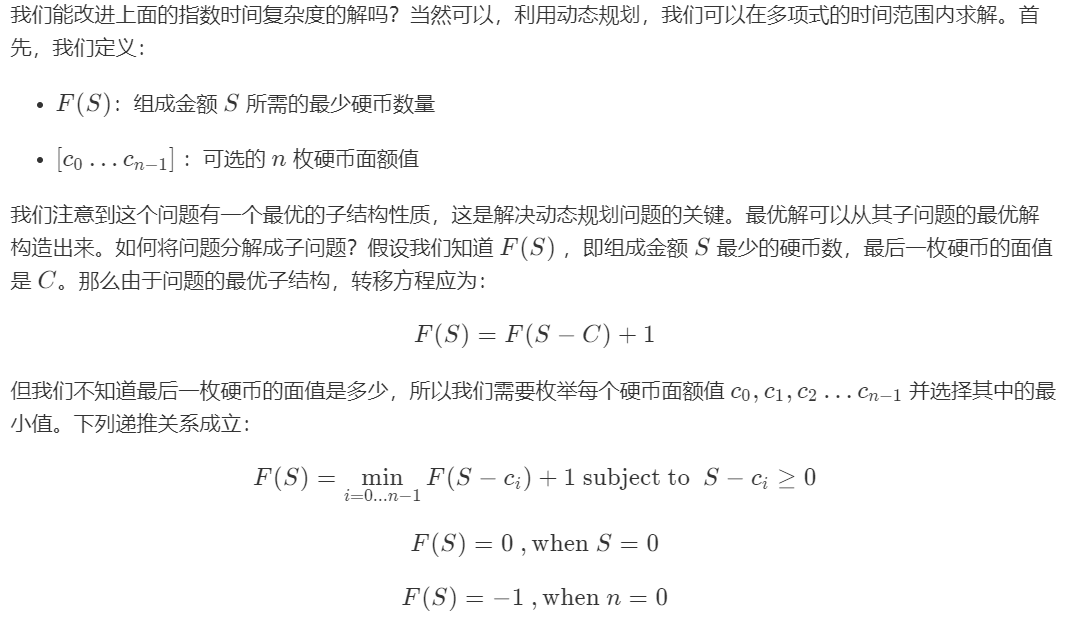
输出: -1

**要求：**

**运用动态规划的思想作答，请写出分析过程和状态转移方程，并用一种语言（最好是C++或JAVA）实现你的思路，代码粘贴在交作业的word里，并保证代码能正确运行，复杂度尽可能低。**

答案：<https://leetcode-cn.com/problems/coin-change/solution/322-ling-qian-dui-huan-by-leetcode-solution/>

方法一：自上而下



方法二：自下而上

