## 四状态定义及转移

例题链接：P714 需要交易费用的股票交易

https://github.com/CyC2018/CS-Notes/blob/master/notes/Leetcode%20%E9%A2%98%E8%A7%A3%20-%20%E5%8A%A8%E6%80%81%E8%A7%84%E5%88%92.md#2-%E9%9C%80%E8%A6%81%E4%BA%A4%E6%98%93%E8%B4%B9%E7%94%A8%E7%9A%84%E8%82%A1%E7%A5%A8%E4%BA%A4%E6%98%93

按CS-Notes/LeetCode题解--动态规划--股票交易中的分析方法（套路）：

1. **四个状态：**

**Sell**----不持股（当天卖出）

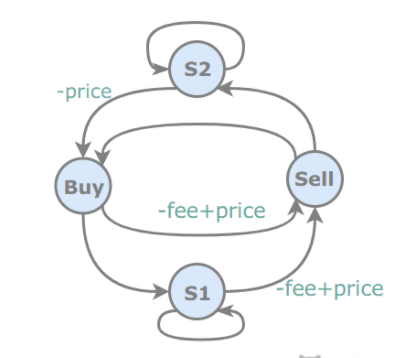
**S2**----不持股（之前卖出）

**Buy**----持股（当天买入）

**S1**----持股（之前买入）

1. **四个状态转移图示：**

注：fee为本题条件—交易费用（不通用）



1. **状态转移**的举例说明：

箭头右侧为第i日，左侧为第i-1日（即一个箭头表示从i-1日转移到i日）

以Buy→S1 及Buy→Sell为例，Buy(i-1)可转移到S1(i)或Sell(i)，解释：

Buy→S1：i日持股，因为i-1日买入

Buy→Sell：i日卖出，因为i-1日买入

以此类推，S1→Sell：i日卖出，因为i-1日持股

## P714带交易费例子分析（状态转移方程+初始条件+结果提取）

Buy[i] = Math.max(s2[i-1], sell[i-1]) – prices[i]

S1[i] = Math.max(s1[i-1], Buy[i-1])

Sell[i] = Math.max(Buy[i-1],S1[i-1]) + prices[i] – fee

S2[i] = Math.max(s2[i-1], Sell[i-1])

初始化边界条件：

S1[0] = Buy[0] = -prices[0]

Sell[0] = s2[0] = 0

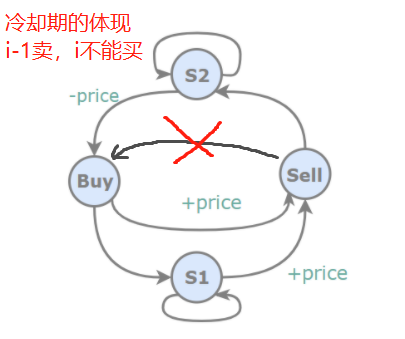
结果提取：

从逻辑上确认，最后一天买入肯定得花钱，收益一定小于卖出

所以最后一天状态应该为不持股（最后一天卖出or最后一天之前卖出）

最大收益 = Math.max(Sell[len-1], S2[len-1])

## P309带冷却期交易分析



四大金刚转移方程：

buy[i] = s2[i - 1] - prices[i];

s1[i] = Math.max(buy[i - 1], s1[i - 1]);

sell[i] = Math.max(buy[i - 1], s1[i - 1]) + prices[i];

s2[i] = Math.max(s2[i - 1], sell[i - 1]);