数据结构与算法

Data Structure and Algorithm

XXVI . 动态规划 III

授课人：Kevin Feng
翻　译：王　落　桐

课前回顾

- 数据结构及算法
- 数学回顾
- 数组（Array）和数组列表（Array List）
- 递归 vs. 迭代
- 二分法搜索
- 分治法
- 链表
- 栈和队列

课前回顾

- 哈希（Hash）
- 树（Tree）
- 堆（Heap）
- 图论（Graph）
- 双向指针（Two Pointers）
- 滑动窗口（Sliding Window）

买卖股票 V

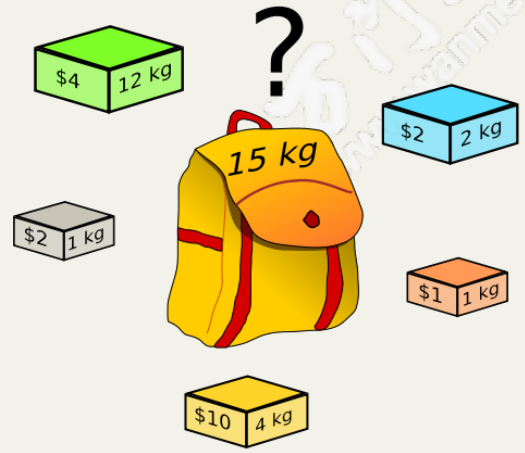
- 给定一个数组，表示每天的股票价格。
- 你可以进行K次的交易（先买再卖，在再次买入的时候必须将之前的股票卖出），问如何能得到最大利润。
- 输入：prices = [3,2,6,5,0,3]， k = 2
- 输出：7

二维动态规划

- 0/1背包问题：0/1 Knapsack
- 最大公共子序列：Longest Common Substring
- 最长增长子序列：Longest Increasing Subsequence

0/1 包裹

- 还记得你曾经打家劫舍吗？为什么你还不知悔改？
- 你背着一个背包闯入一家珠宝店，店里有林林总总的格式珠宝，各不重样。每一个珠宝都有重量和价值。但是你的书包有承载上限。
- 想成为江湖老大，你需要找到一种装包方法使得包内物品的重量不超过其限定值且使包内物品的价值最大
- 这样你的小弟才会对你心服口服！



最长公共子串

- 给定一个序列： $x_1, x_2, x_3, x_4, \ldots, x_n$
- 子序列定义为一串按照顺序的标志，但是并不一定是连续的序列
- 给定两个序列：$X = x_1\, x_2 \ldots x_n$ and $Y = y_1\, y_2 \ldots y_m$
- 最大公共子序列必须满足其同时是X和Y的子序列。我们想要找到最大公共子序列的长度。

最大递增子序列

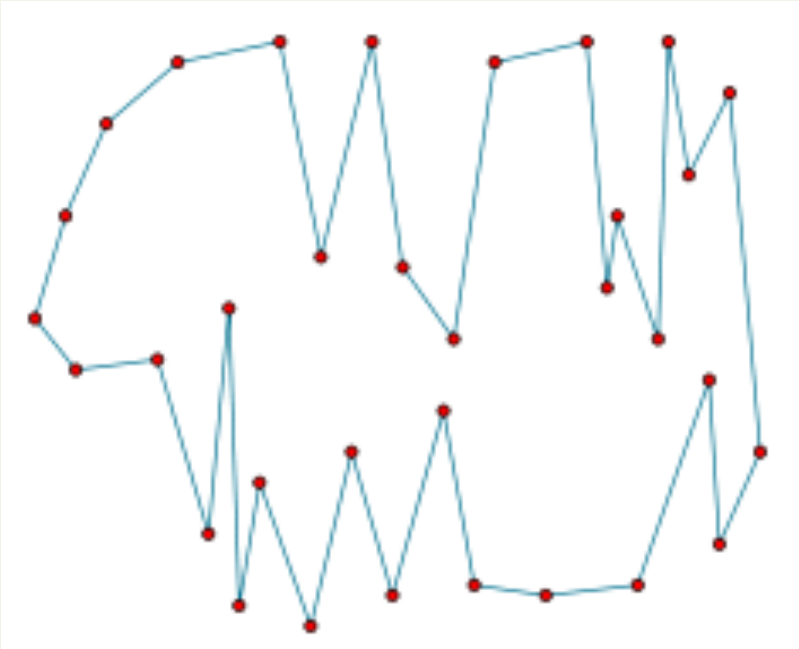
- 给定一个长度为N的数组，找出一个最长的单调自增子序列（LIS）（不一定连续，但是顺序不能乱）。
- 例如：给定一个数组｛10，22，9，33，21，50，41，60，80｝，则其最长的单调递增子序列为{10，22，33，50，60，80}，长度为6.

矩阵链

- A是一个p*q的矩阵，B是q*r的矩阵
- 则矩阵的乘积可以定义为：
- $C = A \times B = \begin{pmatrix} c_{11} & \cdots & c_{1r} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pr} \end{pmatrix}$ is defined by
- $C_{ij} = \sum_{k=1}^{q} a_{ik} b_{kj}$
- 使用标准算法：需要p*q*r次乘法
- 有乘法结合律，$(\boldsymbol{A} \times \boldsymbol{B}) \times \boldsymbol{C} = \boldsymbol{A} \times (\boldsymbol{B} \times \boldsymbol{C})$。然而需要的乘法次数不一定一样。找到最优路径（可以写出括号的可以给出较少的乘次数）

移动销售

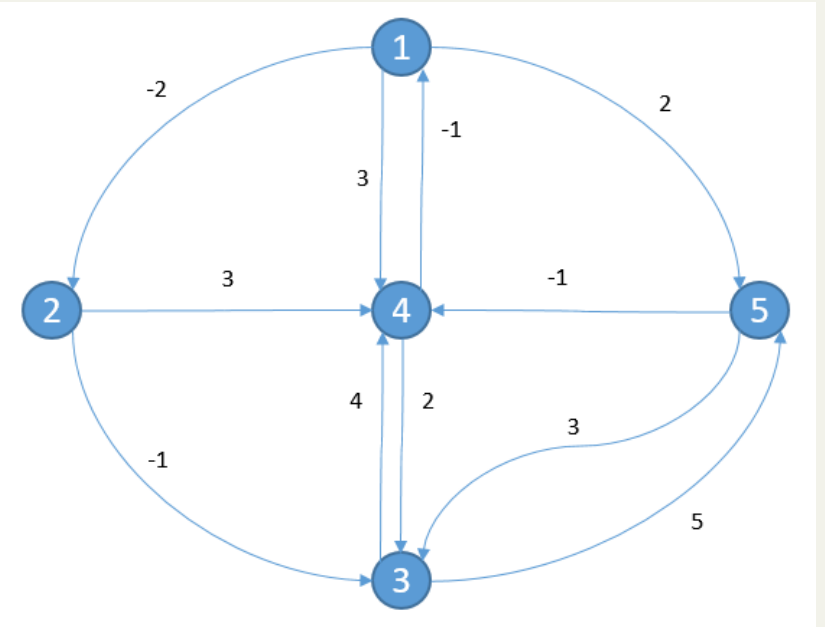
- 在平面上给出n个点，将这些点连成最短的路径（通过欧氏距离测定），要求：从任一点出发，所有点只能访问一次，最终回到起点



Floyd-Warshall 算法

- 我们的目标是寻找从点i到点j的最短路径。

- 从任意节点i到任意节点j的最短路径不外乎2种可能，1是直接从i到j，2是从i经过若干个节点k到j。所以，我们假设Dis(i,j)为节点u到节点v的最短路径的距离，对于每一个节点k，我们检查Dis(i,k) + Dis(k,j) < Dis(i,j)是否成立，如果成立，证明从i到k再到j的路径比i直接到j的路径短，我们便设置Dis(i,j) = Dis(i,k) + Dis(k,j)，这样一来，当我们遍历完所有节点k，Dis(i,j)中记录的便是i到j的最短路径的距离。



数据结构与算法

Data Structure and Algorithm

XXIII. 动态规划 II 结束

授课人：Kevin Feng
翻　译：王　落　桐